For office use only	Team Control Number	For office use only
T1 ________________	8506	F1 ________________
T2 ________________		F2 ________________
T3 ________________	Problem Chosen	F3 ________________
T4 ________________	B	F4 ________________

2018 HiMCM Summary Sheet

As all kinds of machines are getting "smarter", the market demand for a new generation of climate control system is increasing. In this paper, we provided a model for the future generation of smart home climate control systems that can automatically and appropriately adjust the temperature of the house. We firstly built a model for single-zone climate control system. Overall, the model is a constrained multi-objective optimization problem with two main objectives: 1. achieve the user's desired temperatures, 2. reduce the power consumption of the system. In doing so, comfort is maximized while cost is minimized. Its decision variable is the actual room temperature set by the system.

To maximize comfort, the system learns users' desired temperatures at different instants from users' past inputs. As not all historical data are equally predictive of the desired temperature at a particular instant, the system will use autocorrelation analysis to identify periods in users' schedules. The system will then predict the desired temperature based on temperatures set at that time in other periods by taking their weighted average. We then modelled the user's satisfaction as a function of the difference between actual room temperature and the predicted desired temperature. To reduce the electricity bill, we used a thermodynamic model to formulate power consumption as a function of desired room temperature and ambient temperature. Other considerations such as humidity and allergen levels are included in the model as constraints. To optimize these objectives simultaneously, we created a Utility Function that combines the Satisfaction Function and the Power Function. We performed Monte-Carlo Tree Search to find the optimal temperature curve that maximizes the net utility of the users. Our system will set the temperature in the next day according to the optimal temperature curve found.

Furthermore, we adapted our algorithm for a larger house with multiple heating/cooling zones. Multiple-zone systems face the distinct problem of inter-zone thermal interactions, which affects the accuracy of temperature control significantly. We addressed this issue by programming our system to learn the rate of heat transfer between adjacent zones and use this to calculate the power at which the system should function to achieve ideal temperatures. In addition, we discussed the generalization of the satisfaction function to the multiple-residents case.

Finally, we conducted many rounds of performance tests using self-generated data sets. Our performance tests showed that our system can successfully identify periods in users' schedules and its predictions of users' ideal temperatures consistently achieve a high accuracy, with an average absolute error less than 1°C for temperature data sets with different trends, such as long-term temperature shifts and short-term weather anomalies. The tests also showed that our system can learn a new trend typically within two days, minimising required users' manual inputs. Furthermore, our system consistently reduces power consumption by 10% to more than 30%, while keeping the temperature within users' ideal temperature range. In a nutshell, our system has achieved its goal of maximizing comfort while minimizing cost.

Contents

1 News Release **2**

2 Introduction and Problem Restatement **3**
2.1 Introduction 3
2.2 Problem Restatement 3

3 Assumptions and Variables **4**
3.1 Assumptions 4
3.2 Important Variables 4

4 Single-zone Climate Control (Problem 1) **5**
4.1 Model Introduction 5
4.2 Data Collection and Processing 6
4.3 Model (1a) 6
4.3.1 Adapt to Personal Preference 6
4.3.2 Electricity Bill Reduction 11
4.3.3 Obtain the Ideal Temperatures Throughout a Day 15
4.3.4 Additional Considerations 16
4.3.5 Dealing with User Input 18
4.4 Evaluation (1b) 20
4.4.1 Flexibility 20
4.4.2 Learning from Manual Temperature Changes 21
4.4.3 Prediction of users' departure and arrival 22
4.4.4 Energy costs 22
4.4.5 Other geographical conditions (humidity, allergen, and air purity level) . 23

5 Multi-zone Climate Control (Problem 2) **23**
5.1 Introduction 23
5.1.1 Similarities and Differences between Multi-zone Systems and Single-zone System 23
5.2 Learning Parameters 24
5.3 Thermal Interactions between Zones 25
5.4 Conflicting Needs of Users in the same House 26

6 Robustness and Sensitivity Analysis **27**
6.1 Period Identification 27
6.2 Personal Preference - Prediction based on Historical Data 28
6.3 Utility Function 30

7 Strengths and Weaknesses **30**
7.1 Strengths 30
7.2 Weaknesses 31

Appendices **33**

1 News Release

The Future is Here: ThermostaX

X city - SmartX's CEO today introduced the new household climate control systems **ThermostaX** with remarkable learning ability and next-generation performance, marking a tremendous change to household climate control system. The next-generation climate control system integrates an improved sensor system with the most advanced computing chips, adding a breakthrough in automatic temperature controlling and a significant update of energy saving features. This next-generation climate control system has perfectly tackled the problem of limited flexibility and learning capability of current climate control systems, and undoubtedly, it will become an even stronger companion for your family's convenience and health. "Our product is packed with a host of innovative features. The high flexibility of our algorithm enables our product to cope with all kinds of customers' needs." said Dr. Y, SmartX's chief scientist and head of global product development, "No matter what your schedules' periods are, no matter what types of houses you are living in, or how the ambient temperature changes, our smart climate control system can always perfectly find ways to adapt to your preferences, providing you and your family with the most comfortable and convenient user experience with low energy cost."

Learning ability
This new climate control system achieves high performance for automatic temperature controlling. With innovative algorithms, the system can accurately learn from historical user inputs and adapt to users' personal preferences. Rigorous performance tests showed that ThermostaX can predict users' ideal temperatures with an absolute error within 1°C even when the users' temperature preferences are shifting. This demonstrates its amazing precision and adaptibility. Furthermore, this next-generation thermostat offers a core experience of flexibility as it can cope with all kinds of schedules. Unlike its competitors, ThermostaX does not assume anything about its users' schedules and can achieve the same prediction accuracy for users with different schedules. This is indeed a good news for those who do not follow a rigid five-weekdays-and-two-weekend-days schedule.

Energy-saving feature
Sticking to the ideal of "go smart, go green", the new climate control system is not only able to satisfy all our imaginations of an intellectual thermostat, but it also can save up to 30% of energy by adjusting the temperature subtly without affecting users' comfort.

Holistic consideration
SmartX's revolutionized climate control system gives users a new level of comfort. It takes relative humidity, indoor and outdoor temperature difference, and allergens into consideration, and automatically help users modify the room temperature. This means that ThermostaX, while maintaining users' ideal temperature, can also maintain a comfortable humidity and help users to get rid of the allergens in their homes. The system can take better care of its users than themselves!

2 Introduction and Problem Restatement

2.1 Introduction

Nowadays, with the rapid development of economy and technology, people's demands on convenience and comfort of living have increased over the years. With the introduction of household programmable thermostats, scientists are seeking more advanced techniques to make future climate control systems "smarter". Nevertheless, more holistic algorithms are necessary for the development of such climate control systems.

The existing climate control systems allow users to remotely control the heating or cooling system via Apps on their smartphones or in some ways "learn" the users' schedule. For example, the Nest Learning Thermostat[10] developed by Nest company learns the users' preferences and saves energy by adjusting the temperatures accordingly. However, there are limitations for this kind of systems, as their learning algorithms cannot adapt to users' irregular schedules. What we need to do is to make the system more robust so that it takes irregular schedules into account as well.

An ideal climate control system should be able to automatically and appropriately adjust the temperature in its user's house in response to their departures and in anticipation of their arrivals. In face of this challenge, several factors need to be taken into consideration, such as electricity wastage, personal preference, geographical conditions, schedule of the house owners and so on.

2.2 Problem Restatement

Problem 1 We want to build a model for a single-zone climate control system that can perform two main functions: temperature prediction and schedule anticipation.

Firstly, it must be able to predict a schedule of temperature increases and decreases for the customers and automatically adjust the room temperature based on the predictions. The automatic temperature adjustments are made based on electricity consumption and users' personal preferences and schedules, which can be learned in a short period of time. The system should also take into account of ambient factors such as humidity, allergens, and air pollution levels when making temperature adjustments.

Secondly, the system should also respond to users' departures and anticipate users' arrival correctly. For example, it should be turned off when users are away for a long period of time. It should be turned on prior to users' return, so that when they come home the room temperature would be ideal. Additionally, we should consider the irregularity of the customers' schedule, as an irregular schedule makes it hard for the system to predict arrival and users' preferred temperatures.

Problem 2 We also need to address the case of several people living in a large house with many thermostats controlling multiple heating/cooling zones. We need to discuss how these multi-zone control systems faces unique challenges that single-zone control systems do not, such as regulating thermal interactions between rooms. We also need to discuss how, despite the challenges, the algorithm for a single-zone control system can largely be adapted to suit the needs of a multi-zone control system that changes temperature in an entire house simultaneously.

By discussing both cases thoroughly, we can create a more conducive environment for our customers with different needs.

3 Assumptions and Variables

3.1 Assumptions

Assumption: Our users will change the temperature of the system if they feel uncomfortable.
Justification: Most people will not tolerate uncomfortable temperature as their primary purpose for purchasing a climate control system is to increase their comfort level at home.

Assumption: The relationship between a user's satisfaction and the difference between his/her ideal temperature and the actual temperature can be modelled by a normal distribution function.
Justification: The shape of normal distribution function, being bulging in the middle and quickly decreasing at the two ends, is similar to the relationship between satisfaction and the difference between ideal temperature and actual temperature. Though in reality the distribution should be slightly skewed, but since standard deviation (users' ideal temperature range) is usually small, the skewness is negligible.

Assumption: Our system should focus on temperature control.
Justification: The problem statement states that the system's function is to automatically control temperatures and integrate various factors into this control process. Also, many existing smart climate control systems, such as Nest and Honeywell, focus only on temperature control, too.

Assumption: Thermal interactions between two zones can be approximated by one constant, interdependency index, which measures how temperature change in one zone affects the temperature in the other.
Justification: Although in reality how temperature change in one zone affects another depends on a myriad of factors such as the mass of furniture in the latter, normally these factors do not change significantly, so we can regard their collective influence on thermal interaction between two zones as a constant.

Assumption: Adjacency effect exists in the time series of users' ideal room temperatures.
Justification: Adjacency effect, in the context of this problem, is the phenomenon that more recent room temperature data will be more correlated with the users' ideal temperatures for the next day compared to more distant temperature data. This assumption is valid because the probability that the events that may change users' temperature preferences (e.g. transitions of seasons, change of weathers) occur during a time period is lower if that time period is shorter.

3.2 Important Variables

- t is the particular 15-minute time block in a day. $t = 1, 2, 3, \cdots, 96$
- t_{abs} is the duration of absence of the house owner. $t_{abs} > 0$.
- t_{leave} is the 15-minute time block at which the owner leaves the house. $t_{leave} > 0$.

- t_{return} is the 15-minute time block at which the owner returns to the house. $t_{return} > 0$.
- $T_p(i)$ is the predicted temperature for the ith 15-minute time block in a day.
- T_a is the ambient temperature.
- ${T_r}^{[n]}(i)$ is the actual temperature for day n's the ith 15-minute time block. $n = 1, 2, \cdots$. $i = 1, 2, 3, \cdots, 96$
- n is the number of days since the owner started using the system.
- k_c is the thermal conductivity of house wall.
- A is the surface area of walls in thermal contact with the environment.
- V is the volume inside the house.
- $P(T_a, T_r, \dot{T}_r)$ is the power consumed by air conditioning unit to keep rate of change of temperature at $\dot{T}_r$.
- $S(T_r)$ is the user's satisfaction when actual room temperature is at T_r
- $R(\tau)$ is the autocorrelation coefficient, where τ is time lag.
- $w_j^{[n]}(i)$ is the weight for the temperature prediction for the next day's this time block.
- j is the number of period counting from the last period. $j = 1, 2, 3, 4$.
- σ is the standard deviation of the normal distribution used to determine satisfaction function, which is a parameter based on the user's tolerance and willingness to trade ideal temperature for the reduction of electricity bill.
- Q_C is the amount of heat absorbed from the cold reservoir.
- Q_H is the amount of heat ejected to the hot reservoir.
- C is the heat capacity.
- k is interdependency index.
- $Z = \{z_1, z_2, ..., z_n\}$ is the set of heating/cooling zones in a house, $n \geq 2$

4 Single-zone Climate Control (Problem 1)

4.1 Model Introduction

Aim Our aim is to design the algorithm for a climate control system that can automatically adjust room temperature based on users' schedules and personal habits/preferences while taking geographical conditions such as humidity and allergens into consideration.

Method We first developed an algorithm to learn users' preferences based on historical data and used this knowledge to design a satisfaction function that approximates users' satisfaction as a function of actual room temperature. We then formulated a cost function that approximates cost as a function of room temperature and outdoor temperature. By optimizing the combination of satisfaction function and cost function, we obtain the ideal temperature for the next day. Regional conditions can be incorporated into the process by setting them as constraints on possible temperatures.

4.2 Data Collection and Processing

Data Collection The temperature at which the thermostat is set is recorded every 15 minutes. All the data from the days within the most recent 2 months will be stored in the system's database.

Form of the Data We discretized each day into $\left(\frac{60}{15}\right) \times 24 = 96$ 15-minutes time blocks. In a typical day, the first 15 minutes (00:00-00:15) is the 1_{st} 15 minutes, while the last 15 minutes (23:45-00:00) is the 96_{th} 15 minutes. Therefore, each day's data – a discretized time series – consists of 96 data points. We have two databases. The first database helps us detect periodicity in the user's schedule. It stores the temperatures set when it is on, and stores a large negative value (-99999) when it is turned off in case of user's prolonged absence. The second database helps us predict the ideal temperature and stores the most possible temperature for specific 15-minutes time blocks. We use the term "the most possible temperature" because the missing data when a user is absent is extrapolated from the temperature data of adjacent time blocks. In this manner, we obtain 96 temperature data points in a day.

4.3 Model (1a)

4.3.1 Adapt to Personal Preference

Personal preferences, influenced by one's habits and schedules, are a key factor when deciding when and how the thermostat should change or maintain its temperature settings. We can learn our users' personal preferences based on historical temperature data, with the assumption that they will change the temperature whenever the room temperature deviates from their preferred temperature. Once we have learnt a user's preferences, we can predict their desired temperature in a specific 15-minute time block for the next day.

Periodicity To help improve the accuracy of our predictions, we factored in the periodicity in users' schedules when learning preferences. In real life, our schedules are usually periodic. One example is that we sleep for a fixed number of hours everyday. Another example is that many of us go to work for five days then rest for two days. This means that an office worker's weekly schedule is likely similar throughout the year. The elderly's schedule, on the other hand, may repeat itself after just one day, as they do not go to work and their schedule for everyday is largely the same. It is important to base our predictions, both for preferred temperatures and for duration of absence, on such periodicity because combining data from different days within a period to generate predictions for a new day will lead to high inaccuracies. For example, let us consider a couple, who are both doctors, work for 9 days and rest for 5 days in two weeks. Their elderly parents are the sole occupants of the house when the couple are working. When they are working, their parents leave the house for a short while to do shopping (2 hours). And on their resting days they go hiking with their parents to countryside for a long time (8 hours). In this case, using data from both their workdays and resting days to predict the temperature change and their likely duration of absence on a resting day will surely lead to inaccuracies. The accuracy is improved if we identify the period of their schedule and use data from days in the same position within a period (in phase) to predict the user's ideal temperatures and duration of absence next day. Therefore, periodicity identification is both useful and necessary.

Autocorrelation is the correlation between observations of the same object as a function of the time lag between them. It is a mathematical tool for finding repeating patterns[19]. It is measured by autocorrelation coefficient ($R(\tau)$), which can be mathematically expressed as:

$$R(\tau) = \frac{\mathrm{E}[(X_t - \mu)(X_{t+\tau} - \mu)]}{\sigma^2} \tag{1}$$

where τ is the time lag, μ is the mean of the series and σ is its variance. In the context of this problem, the series will be the the temperatures at which the thermostat was set in the past two months. If $\tau = 1$, then the time lag is one time block, that is, 15 minutes. Natural minimum periods of a person's schedule ranges from one day to about a month, so $\tau = 1 \times 96 = 96, 2 \times 96 = 192, 3 \times 96 = 288, \cdots, 31 \times 96 = 2976$ are the values to consider. If the day we are to predict is the $\mathrm{n_{th}}$ day, the series will be:

$$T_r^{[n-62]}(1), T_r^{[n-62]}(2), \cdots, T_r{}^{[n-1]}(95), T_r{}^{[n-1]}(96) \tag{2}$$

Applying correlation analysis on the series for 31 different τ values, we can get the 31 R values (equation 1). The R values that are above 0.7 (a significant correlation) are chosen, and among them, the one that corresponds to the smallest τ is chosen. This is because the shorter the period, the more recent the historical data used for prediction is, and according to adjacency effect, the more accurate our prediction will be. Our tests largely confirmed this assertion (see Appendix B). The corresponding τ of this value is then regarded as the period of the person's schedule. This test for periodicity is carried out each day to ensure that the system can quickly learn changes in the period such as when school holiday begins.

Temperature Prediction We can then calculate the expected temperature at a specific 15-minute time block for the next day, using historical data that is in phase with the day in interest. The system considers the historical data in five most recent periods. For example, suppose we want to predict the temperature at the 1st time block in day 65 ($T_r^{[65]}(1)$), and the identified period of the user's schedule is 7 days. We will use the data points from the 1st time block in day 58, 51, 44, and 37 ($T_r^{[65-7\times1]}(1), \quad T_r^{[65-7\times2]}(1), \cdots$) to predict it. We predict temperature at different time blocks separately, considering that it is very possible that only a part of a day's schedule changes.

The expected temperature at a specific 15-minute time block in the next day can be predicted as the weighted average of the past data. The weights are assigned according to each data point's predictive power. Suppose we are predicting the $\mathrm{i_{th}}$ 15-minute time block on the $\mathrm{(n)_{th}}$ day. The actual temperature (unknown for now), is $Y = T_r^{[n]}(i)$. For the ease of reading, all (i) in the section will be omitted, and all the variables below are specific the $\mathrm{i_{th}}$ time block. This user's period is τ (in days). The data points used for this prediction are:

$$X_{predict}^{[n]} = (T_r^{[n-\tau]}, T_r^{[n-2\tau]}, T_r^{[n-3\tau]}, T_r^{[n-4\tau]}), T_r^{[n-5\tau]}) \tag{3}$$

We can approximate the predictive power of the last four points

$$X^{[n]} = (T_r^{[n-2\tau]}, T_r^{[n-3\tau]}, T_r^{[n-4\tau]} T_r^{[n-5\tau]})$$

by

1. considering their absolute differences from the first data point ($T_r^{[n-\tau]}$), which is the most recent data point
$$X_{dif}^{[n]} = |T_r^{[n-\tau]} - X^{[n]}|^* \tag{4}$$
*The terms of $X_{dif}^{[n]}$ are $|T_r^{[n-\tau]} - T_r^{[n-2\tau]}|$, $|T_r^{[n-\tau]} - T_r^{[n-3\tau]}|$, and so on...(matrix broadcasting)

2. negating these differences in a certain way so that the data that is closer to the most recent data point is assigned a larger index.

$$M = \max X_{dif}^{[n]} \tag{5}$$

$$X_{power}^{[n]} = (M + 0.001) - X_{dif}^{[n]*} \tag{6}$$

*Again using matrix broadcasting as in **1.**

Each value in $X_{predict}^{[n]}$, a positive value, is a measure of the predictive power of the corresponding value (in the same position) in X, as we assume that if a historical record is closer to the most recent temperature data (from the same position within a period as the day to be predicted), then it has a greater predictive power.

3. Normalizing this index to obtain the weight vector.

$$W^{[n]} = (w_1^{[n]}, w_2^{[n]}, w_3^{[n]}, w_4^{[n]}) \tag{7}$$

where

$$W^{[n]} = \frac{X_{power}^{[n]}}{\sum X_{power}^{[n]}} \tag{8}$$

4. As we are also using $T_r^{[n-\tau]}$ as a historical data for prediction, we need to add an additional weight to the weight vector. Assuming adjacency effect holds, we set this additional weight for $T_r^{[n-\tau]}$ as the same as the weight for $T_r^{[n-2\tau]}$, $w_1^{[n]}$. And normalize the weights again as in **3.**

$$W^{[n]} = (w_1^{[n]}, w_1^{[n]}, w_2^{[n]}, w_3^{[n]}, w_4^{[n]}) \tag{9}$$

5. However, if the temperatures show a trend of consistently increasing or decreasing, as it is usually in the case during transitions between seasons (such as summer to autumn, people naturally wear more indoor) we cannot simply use these five weights which sum up to 1. Thus, an additional adjusting term is subtracted from all the weights if $T_r^{[n-\tau]}$ $< min\ X^{[n]}$ and added to all the weights if $T_r^{[n-\tau]} > max\ X^{[n]}$, ensuring the trend can be accurately predicted. The adjusting term is:

$$\frac{\min X_{dif}^{[n]}}{\sum X_{predict}^{[n]}} \tag{10}$$

The weights for historical data are derived. The predicted ideal temperature on day n in i$_{th}$ time block is

$$T_p^{[n]} = W^{[n]} \cdot (X_{predict}^{[n]})^{\mathrm{T}} \tag{11}$$

Lastly, up to now we did not use temperature data from the days close to "today" (the day to be predicted) as they are from different positions in a period. However, if the recent days' data shows a systematic drift from previous data, as in the case when cold wave suddenly hit the city two days ago, then we should add this systematic drift to our predicted value. Thus, the final predicted ideal temperature on day n in i$_{th}$ time block is

$$T_p^{[n]} = W^{[n]} \cdot (X_{predict}^{[n]})^{\mathrm{T}} + \delta \tag{12}$$

where δ is the average difference between temperatures from the most recent τ days/period and their corresponding temperatures (in the same position in a period)from the second most recent τ days/period.

In this way, we can predict the temperature for all 96 time blocks in the next day using historical data, taking into consideration both information from closer dates and information from more distant dates. This prediction method has a high adaptability and can easily adapt to long term changes (e.g. transition between seasons) and short term anomalies (e.g. a short-term cold weather) in temperature data. It is also highly flexible as it can predict temperatures for people with all types of schedules and routines as it can learn different types of periods. It does not assume its users to have a "five weekdays and two weekend days" schedule. Even if one's work demands one to only rest for three days in a month, it can adapt to this schedule well (More details will be provided in the **Robustness and Sensitivity Analysis** section).

Starting the System When a user first install the system in his house, the system does not have enough historical temperature data. To start the system:

1. In the first two days the user needs to set the temperatures manually and change it according to his preferences.
2. Starting from the third day, the algorithm will begin to test for periodicity.
   (a) If historical temperature data for the first two days are similar ($R(1) > 0.7$), the algorithm will set the period as one and predict the third day's temperature from the two.
   (b) If they are not, then the algorithm will continue to learn new manual inputs while trying to detect periodicity.
3. If no period can be identified after one week. The algorithm will automatically set the period as one week and use the data from one period to predict. In the next three weeks, the algorithm will continue trying to detect period and if any better period is found than "one week", the former will replace the latter.
4. The algorithm uses less than four periods of data to predict the temperature for period 2, 3, and 4, but the method, weighted average (please refer to **Temperature Prediction** part above), is still the same. Below is a flowchart demonstrating the starting procedures of our system.

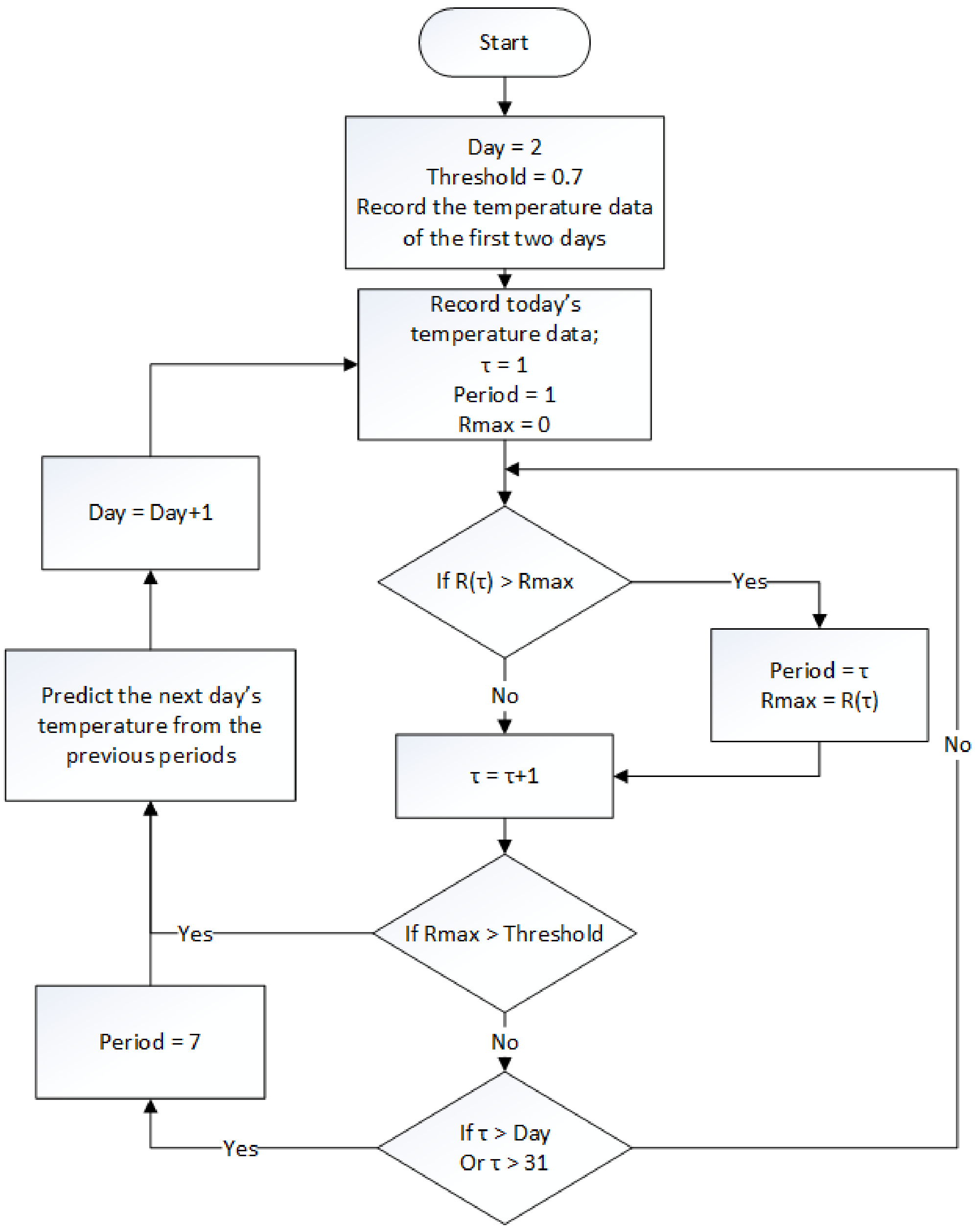

Figure 4.1. A flowchart demonstrating the starting procedures of our system.

Satisfaction Function In the above paragraphs, we discussed how to predict the ideal temperature of the next day based on user's habits. However, the user's ideal temperature is not likely to be a specific value, but a range of values. Indeed, we can calculate a satisfaction function ($S(T_r, T_p)$ which describes the user's satisfaction as a function of the room temperature and ideal temperature at that time (which reflects his preference). According to studies [1] [20], humans' comfort temperature typically has a 1°C tolerance range. But according to another study [2], humans are very sensitive to temperature differences (is able to detect a difference around 0.1°C). To accommodate different tolerances, we can model the satisfaction as a normal distribution with a mean equal to ideal temperature.

$$S(T_r, T_p) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\dfrac{(T_r - T_p)^2}{2\sigma^2}} \tag{13}$$

σ, the standard deviation, is a parameter based on the user's tolerance and willingness to trade ideal temperature for the reduction of utility bill. σ^2 ranges from 0.5 (almost intolerant) to 3.5

(high tolerance). The system asks for user input through a slider (0-5) and scale the input to get the value of σ^2: $\sigma^2 = (\frac{3}{5}) \times \text{input} + 0.5$

4.3.2 Electricity Bill Reduction

Another important consideration in designing an intelligent climate control system is its power consumption. Ideally, the power consumption of the system should be kept to a minimum while maintaining comfort of the house owner. The power consumption of the system is a function of the desired room temperature, the ambient temperature, and the derivative of room temperature.

$$P = P(T_r, T_a, \dot{T}_r) \tag{14}$$

Before we begin to calculate the required power to keep the temperature on track, we must first discuss about the efficiency of the air conditioning unit. This is best done with the use of the Coefficient of Performance (COP)[14], which is the amount of heat transferred from indoors to outdoors per unit of electrical energy used. Mathematically, the COP can be expressed as

$$\text{COP} = \frac{Q_C}{W} = \frac{Q_C}{Q_H - Q_C} \tag{15}$$

where Q_C and Q_H are the amount of heat absorbed from the cold reservoir and ejected to the hot reservoir respectively. In the second equality we have used the first law of thermodynamics, which states that the total energy in the system must be conserved.

A real refrigerator (and hence an air conditioner) runs the Rankine cycle in reverse[14], with the help of a throttling valve. The Pressure-Volume diagram is shown below. The red process is isobaric cooling in the evaporator, the green process is an adiabatic compression by the compressor, the orange process is isobaric condensing in the condenser, and the cyan process is thermal throttling.

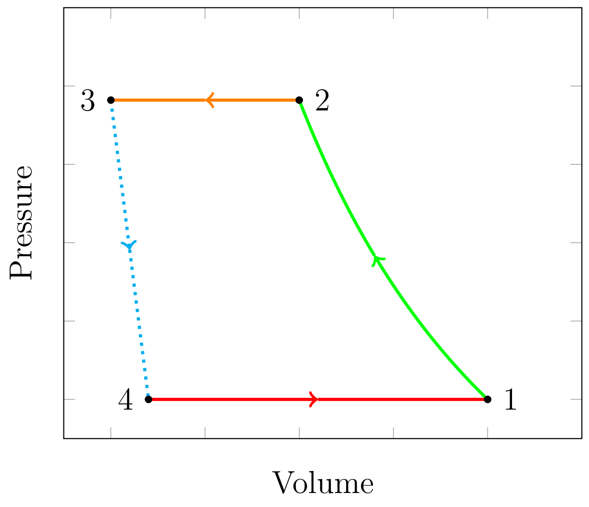

Figure 4.2. The Pressure—Volume diagram of a reversed Rankine cycle.

From the PV diagram, it is easy to see that Q_C is the heat absorbed in the red isobaric (*i.e.* constant pressure) process in the evaporator and that Q_H is the heat ejected in the orange isobaric process in the condenser. Since these two processes are isobaric, the amount of heat absorbed is equal to the enthalpy change of the refrigerant. Enthalpy is defined as

$$H = U + PV \tag{16}$$

and should be interpreted as the energy required to create something and put it into its surroundings, which requires work to be done. Enthalpy is also a state function and is independent of the thermodynamic process undertaken to move from state A to state B. To see why enthalpy change is equal to heat exchanged in an isobaric process, we take its total differential and invoke the first law of thermodynamics again,

$$\mathrm{d}H = \mathrm{d}U + P\,\mathrm{d}V + V\,\mathrm{d}P = dQ + V\,\mathrm{d}P = dQ \tag{17}$$

since $\mathrm{d}P = 0$ for an isobaric process. Therefore, the COP can be expressed in terms of enthalpy as

$$\mathrm{COP} = \frac{H_1 - H_4}{H_2 - H_3 - H_1 + H_4} \tag{18}$$

Note that $Q_H = H_2 - H_3$ since heat *left* the system. Now, it is well known that throttling does not change the system's enthalpy, so that $H_3 = H_4$. Thus, the COP simplifies to

$$\mathrm{COP} = \frac{H_1 - H_3}{H_2 - H_1} \tag{19}$$

The enthalpy at the three states can be read from tables of thermodynamic properties of saturated liquid/vapor and superheated vapor of refrigerant. A common refrigerant currently in use is HFC-134a, and guidelines indicate that the air conditioner should operate over pressure ranges of between 40 psig ($\approx$ 3.5 atm) and 150 psig ($\approx$ 11 atm). We can determine state 2 by insisting that the reversible adiabatic process does not change the system's entropy, so that $S_1 = S_2$. The following data are taken from the table[6]:

State	Temperature(°C)	Pressure(kPa)	Enthalpy(kJ/kg)	Entropy(kJ/kg·K)
1	5	349.87	401.7	1.7252
2	48	1100.00	425.5	1.7252
3	43	1101.93	261.1	1.2053

Table 4.1. Thermodynamic Properties of HFC-134a

This gives a theoretical COP of about 5.9. In practice the COP will be lower due to a variety of imperfections (such as electrical resistance), but COP $\approx$ 4 will suffice for our purposes. It should also be mentioned that the COP calculated above is properly called the COP of cooling. If we want to heat up the house instead, we will be interested in COP of heating, which is given by

$$\mathrm{COP}_{\mathrm{heat}} = \frac{Q_H}{Q_H - Q_C} = \mathrm{COP}_{\mathrm{cool}} + 1 \tag{20}$$

The next important thing to consider is, when will the air conditioning unit consume power? Imagine for a moment that the house is completely isolated from the ambient environment. Then there would be no point of turning on the air conditioner apart from the initial cooling. We shall calculate how much energy that takes.

To calculate the amount of energy is needed to bring down the temperature, we need to find out the combined heat capacity of things inside the house. We shall consider three types of substance inside the house: air, furniture, concrete walls.

Firstly, air at common room conditions (~25°C, 1 atm) can be modeled as a diatomic ideal gas[17]. Hence, the heat capacity of air at constant volume is given as

$$C_V = \frac{5}{2}nR \tag{21}$$

Using the ideal gas law $PV = nRT$, we can write the heat capacity of air as

$$C_V = \frac{5}{2}\frac{PV}{T} = \frac{5}{2}\frac{PSh}{T} \tag{22}$$

where S is the floor area of the house and h is the height. Since the house we are considering is small, we take $S = 30$ m^2 and standard floor height $h = 3$ m. With $P = 101000$ Pa and $T = 298$ K, we find that $C_V \approx 76300$ J/K.

Secondly, we look at the heat capacity of furniture in the house. Wood has a specific heat capacity of about 1.7 J/g·K and aluminum has specific heat capacity of about 0.9 J/g·K. An estimate of total mass of furniture in the small apartment can be around 50 kg, and hence the total heat capacity is also about 70000 J/K.

Lastly, we look at the heat capacity of walls[18], ceiling, and floor surrounding the house. We take these surfaces to be made of concrete, and consider only half of their thickness (since the other half is shared with other apartments). The density of normal concrete is reported to be 2400 kg/m^3 and the specific heat capacity is 0.880 J/g·K. Let us take the dimensions of the house to be 6m×5m×3m, total surface area is 126 m^2 and hence total concrete volume is 6.3 m^3. This gives a heat capacity of 1.33×10^7 J/K, dwarfing everything inside the room. In fact, the concrete surrounding the building can store so much energy that it is considered a good thermal mass. It indicates that we should really be thinking about the system as consisting of three components: the room, the wall, and the environment (heat reservoir). Look at the following diagram.

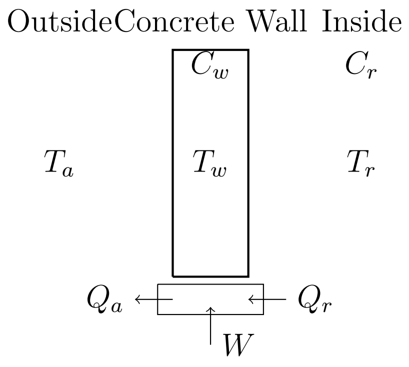

Figure 4.3. A diagram demonstrating the relationship between the room, the wall and the environment.

The wall exchanges heat with both the environment and the room through direct conduction. Additionally, the room exchanges heat with the environment with the help of the air conditioner. Some electrical work needs to be done by the air conditioner, and this is the power of the AC unit we are looking for. It is easy to write down the coupled differential equations describing the temperature of the wall and the room. We assume that the temperature of the environment is a given function, obtained from past meteorological data. Assuming that volume change due to heating and cooling is negligible, the energy change in a system is just

$$\Delta U = C\Delta T = \Delta Q \tag{23}$$

Therefore, the temperatures of all three components are given by

$$T_a = T_a(t) \tag{24}$$

$$C_w\dot{T}_w = \frac{kA}{L/2}\left[(T_a - T_w) + (T_r - T_w)\right] \tag{25}$$

$$C_r\dot{T}_r = \frac{kA}{L/2}(T_w - T_r) - Q_r \tag{26}$$

where k is thermal conductivity of the wall, A is the surface area of the wall, and L is the thickness. Rearranging equation (26) to make Q_r the subject, we find that

$$Q_r = \frac{kA}{L/2}(T_w - T_r) - C_r\dot{T}_r \tag{27}$$

Q_r is the heat absorbed from the room, so if the room was heated instead, Q_r would be negative. This means that the power drawn by the AC unit is

$$P(T_r, T_a, \dot{T}_r) = \begin{cases} [2kA(T_w - T_r)/L - C_r\dot{T}_r]/\mathrm{COP}_{\mathrm{cool}}, & Q_r \geq 0 \\ [C_r\dot{T}_r - 2kA(T_w - T_r)/L]/\mathrm{COP}_{\mathrm{heat}}, & Q_r < 0 \end{cases} \tag{28}$$

This function is not differentiable at its corners, where $Q_r = 0$. This means that we cannot use the Euler-Lagrange equation to minimize it[5]. And rather unfortunately, its partial derivative with respect to $\dot{T}_r$ is not continuous, so we cannot use the generalized Euler-Lagrange equation for piecewise smooth functions either. Discretization comes into play here and we can numerically optimize to obtain the best temperature curve.

It should also be noted that it would be very difficult for a machine to learn the power consumption of the air conditioning unit due to the number of different factors involved. For example, the weather would have a big impact on the actual power consumption of the system, as solar radiation adds much heat and rain evaporation takes away much heat. Also, the temperature in your neighbors' rooms also affect how fast heat is flowing into or out from your room. These large variations means it is difficult for a computer to learn the relation between temperature differences and the power consumed. By using this manual method, we are not disregarding these other factors. Rather, we are arguing that in long periods of time the effects of these factors are usually averaged out. There are roughly the same number of rainy days in a particular month in different years, for instance. Nevertheless, we do acknowledge that these variations cause a large variation on a daily basis, as elaborated in the next section.

Effect of Solar Radiation It is worthwhile to include a discussion on the impact solar radiation has on the temperatures. In practice, the impact solar radiation has on the wall is often very significant. Concrete has an albedo of about 0.5, meaning that it reflects about 50% of the light shining upon it. Solar constant can be taken to be around $1000\,\mathrm{W\,m^{-2}}$. Assuming the area of the irradiated surface is $30\,\mathrm{m^2}$, power of solar radiation is about 1.5×10^4 W, roughly equal to conduction with a temperature difference of 15°C.

However, solar radiation rarely directly hits the concrete wall. The amount of energy incident upon the wall thus depend on many other factors, such as the weather and the altitude/azimuth of the sun on the celestial sphere. These effects combine into the optical depth of the atmosphere, which is in general a quantity that is difficult to predict due to the sheer number of factors that influences it. The solar constant just outside Earth's atmosphere is roughly $1383\,\mathrm{W\,m^{-2}}$, but at most 75% of it ($\approx 1000\,\mathrm{W\,m^{-2}}$) reaches the ground. The percentage of radiation that makes its way to the surface is given by the Beer-Lambert law[4], which, for our intents and purposes, states that

$$\frac{I_f}{I_i} = e^{-m(\tau_a + \tau_g + \tau_{\mathrm{RS}} + \tau_{\mathrm{NO_2}} + \tau_{\mathrm{H_2O}} + \tau_{\mathrm{O_3}} + \tau_r + \cdots)} = e^{-m\tau} \tag{29}$$

The m is called relative air mass and τ is the optical depth of the atmosphere. If we assume a parallel plane atmosphere, m is given by $1/\cos\Phi$, where Φ is the zenith angle of the sun. This means that solar power on the surface is roughly

$$I \approx 1383 e^{-\tau/\cos\Phi} \tag{30}$$

in SI units. τ usually ranges from 0.3 to 2.3, depending on the conditions at hand. It could be measured by using a solar photometer to take multiple readings and regression analysis, but for our purposes we will just take solar radiation as a directly measurable quantity.

Another consequence of the variance in altitude of the sun is that sunlight is not always directly incident upon the wall. Instead, the area of the wall should be multiplied by a factor of $\sin\Phi$ since we are interested in the area perpendicular to the direction the rays are coming from; *i.e.* we are interested in the flux through the wall.

Nevertheless, if we are to take solar radiation into account, we must also consider the thermal radiation emitted by the room and the wall. Concrete has an emissivity of 0.94[11], so if we continue to assume an area of $30\,\mathrm{m}^2$, according to the Stefan-Boltzmann law for black body radiation

$$I = \sigma\varepsilon A T^4, \tag{31}$$

at room temperature the wall can give off as much as 1.2×10^4 W to the ambient environment and the room itself. For air the story is much different, and the gases do not lend themselves to a simple equation that gives the amount of thermal radiation they emit. Overall, due to the number of variations that could affect radiation-related heat transfers and the intractability of them computationally, we drop these terms in our predictions of the ideal temperature in the room.

4.3.3 Obtain the Ideal Temperatures Throughout a Day

Utility Function The objective of our climate control system is two-fold: both to maximize the user's satisfaction and to minimize the power consumed by the system. Hence, we decided to take a weighted sum of the two objectives as the functional to be minimized. We define the *utility function*, given by

$$U = \alpha S + \beta P \tag{32}$$

where S is the satisfaction function and the P is the power consumed. It should be obvious that $\beta < 0$ as greater power consumption is less desirable. Typically, α should be at least a few thousand times larger than β to make the two functions roughly equal in magnitude. As established previously, methods for continuous curves cannot be employed to maximize this functional due to its discontinuous nature, so we must seek numerical methods instead.

Nevertheless, a brute force method is also not feasible, due to the shear number of possibilities involved. Suppose that we allow temperature to vary at step sizes of 0.5°C over the range between 16°C and 35°C. There are 39 possible temperatures for each time block, and the corresponding number of possible temperature curves over the day is huge: $39^{96} \approx 5.5 \times 10^{152}$. This is clearly computationally intractable. What if we instead restrict the temperature variation between adjacent 15 minute blocks to be within 2°C? In such a tree search, the number of possibilities become $9^{96} \approx 4.0 \times 10^{91}$. That is a reduction by a factor of 10^{60}, but this number is still huge - about 10 billion times the number of atoms in the observable universe.

Monte-Carlo Tree Search Fortunately, 10^{91} is not large compared to the number of possible games of chess or games of go. Moreover, the climate control system is similar to a game of chess in the sense that the optimal strategy does not depend on previous actions. Previous decisions only determine the starting conditions at a particular time but does not affect the way in which decisions should be made from this point onward. As a result, we drew inspirations from the algorithms that are able to perform at superhuman levels in games of go, namely the Monte-Carlo Tree Search (MCTS) algorithm. It is less clear whether a traditional min-max algorithm with alpha-beta pruning would still work, since there lacks another agent to work against.

In MCTS with a given starting state, the games is played out at random until the end. The result is back propagated upwards until the starting state is reached. The next state to explore is chosen based on how well it performed so far and how unexplored it is. This processes is repeated until computational resources are exhausted, at which point the most promising move is selected. In our specific case, the outcome of the game is taken to be the sum (or integral) of the utility function. We can incorporate various constraints on the system by limiting the legal moves at each state; the details of these constraints are presented in the next section.

With user preferred temperature equals 23°C at all times, standard deviation 0.7°C, and an initial room and wall temperature equal 26°C, the algorithm achieves in ten runs an average reduction in energy cost of 14.6% and standard error of 2.58%. Raw data will also be presented in the appendix. The number of simulations run at each time block is 100,000. Admittedly, more data could be taken to justify the application of the central limit theorem, but the task is still computationally intense and could not be readily performed. With higher number of simulations the reduction in energy cost is more significant, but with a slower computer 1,000,000 simulations could easily take up nearly 10 minutes of CPU time. While the implementation of the algorithm could be optimized to reduce time and memory consumption, we take this as a proof of concept that the MCTS algorithm is able to meet our needs of maximizing comfort while minimizing energy cost.

4.3.4 Additional Considerations

Ambient Temperature According to scientific studies[8], the sudden change in temperature from extreme hot to cold can have a serious effect on people who have cold-related disorders. It can also exacerbate coronary heart diseases, vascular cardiac diseases, vascular brain diseases, and peripheral vascular diseases. Studies [16] show that indoor and outdoor temperature difference should not be greater than 10°C in prevention of respiratory disorders. And if the customer has cold-related disorders, the temperature difference should be even smaller. Since different people have different resistance towards the sudden change in temperature, we allow users to change the default setting of 10°C according to their needs. Once this range is set, when the predicting ideal room temperatures, any solution not inline with the requirements are outright rejected.

Humidity In real life, relative humidity influences people's comfort level significantly. Relative humidity is a more accurate measure of human comfort[1] than absolute humidity as, unlike the former, the latter does not take temperature into consideration. When absolute humidity is the same, humans may still experience different levels of comfort as temperature changes. Therefore, we design the system so that it can maintain the indoor relative humidity in a certain range that makes our users the most comfortable.

In this model, we assume that we cannot directly control the humidity via the climate control system as the question only states that the system can adjust temperature. So we can only control relative humidity by adjusting temperature according to the indoor actual vapor density (H_a) sensed by a sensor. Relative humidity (RH) can be calculated as[7]:

$$RH = \frac{H_a}{H_s} \tag{33}$$

H_s is the saturated vapor density, the only variable that can be controlled by the system. H_a is the humidity sensor's reading at that instant and RH is a range of relative humidity the user is comfortable with (default: $RH_{min} = 0.3$, $RH_{max} = 0.7$).We can rewrite the equation as

$$\frac{H_a}{RH_{max}} \leq H_s \leq \frac{H_a}{RH_{min}} \tag{34}$$

H_s's relationship with indoor temperature is as follows (based on The Buck Equation[13]):

$$H_s = \frac{R(T+273)}{PM} = (\frac{R(T+273)}{M})(0.61121\exp((18.678 - \frac{234.5}{T})(\frac{257.14+T}{T}))) \tag{35}$$

where P is saturated vapor pressure, $R = 8.314(J/mol)$ is the gas constant and $M = 18.0(g/mol)$, is the molar mass of water and T is temperature in Celsius. H_s increases monotonously as T increases. Using these equations, we can obtain the boundary for temperature (T) – T_{max},T_{min}, given H_a, RH_{max}, and RH_{min}. When the predicted ideal temperature $T_p \notin [T_{min}, T_{max}]$, the system can automatically set it to the nearest boundary to maintain a comfortable relative humidity.

Allergens Allergy is another important consideration. The influence of some allergens is closely tied with relative humidity and temperature in the surroundings. Such allergens include dust mites[15], pollen[3], mold[9], etc. Our system keeps a database of these allergens as well as information about the temperature and relative humidity range which maximizes their influence. A sample section of our database is shown below:

Allergen	Optimum Temperature/°C	Optimum Relative Humidity
Dust Mites	20 - 25	0.7 - 0.8
Mold	15 - 30	0.7 - 0.9
Asthma	10 and below	0.6 and above
Pollen	21 - 30	0.3 - 0.5

Table 4.2 A Sample Section of Our System's Allergy Database

When a user begins to use the system, the system asks the user to input the allergies they have and retrieve their information from the database. When the room temperature range falls within the optimum temperature range, the system will regulate the relative humidity using the same method outlined in **Humidity** section above to try its best to make sure the humidity level is not within the optimum relative humidity range for that allergen. This feature ensures that users' allergies are much less likely to be triggered indoor.

Below is a flow chart demonstrating the entire algorithm we used to predict the ideal temperature throughout the day while taking user's personal preferences and schedules, power consumption and regional conditions (ambient temperature, humidity, and allergens) into considerations.

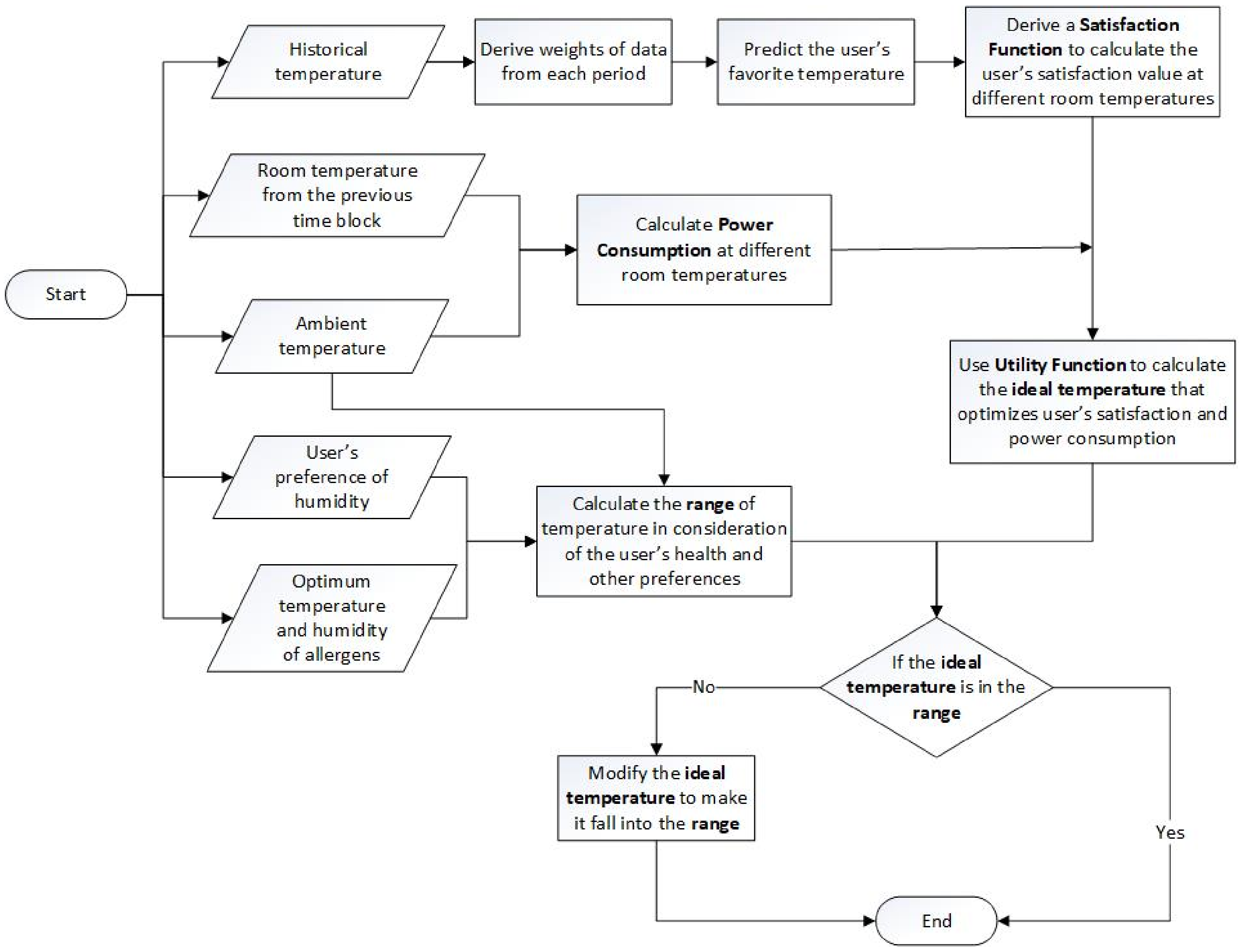
Start
Historical temperature
Derive weights of data from each period
Predict the user's favorite temperature
Derive a Satisfaction Function to calculate the user's satisfaction value at different room temperatures
Room temperature from the previous time block
Calculate Power Consumption at different room temperatures
Ambient temperature
Use Utility Function to calculate the ideal temperature that optimizes user's satisfaction and power consumption
User's preference of humidity
Optimum temperature and humidity of allergens
Calculate the range of temperature in consideration of the user's health and other preferences
If the ideal temperature is in the range
No
Yes
Modify the ideal temperature to make it fall into the range
End

Figure 4.4. A flowchart demonstrating the entire algorithm we used to predict the ideal temperature throughout a day with different factors taken into consideration.

4.3.5 Dealing with User Input

In the above section, we have designed a mathematical model to predict the ideal temperature throughout the next day accurately based on historical data. However, the user may not be completely satisfied with the calculated ideal temperature due to some reasons (*e.g.* sudden weather change, etc.) and would like to manually adjust it. Or he may not be at home all the time, so the thermostat should automatically turn on and off when he leaves or returns.

Manual Adjustment of Temperature When the user manually adjust the temperature at which the thermostat is set so that it now deviates from the predicted ideal temperature, our system responds by adding $T_{\text{set}} - T_{\text{predict}}$ to all the calculated ideal temperature afterwards on that day. On the next day, should the user wish to maintain the adjustment, he would need to adjust the temperature again. However, after another day our system will learn that the user wish to maintain the adjustment (see equation 12) and the weights will be automatically adjusted. Thus, consistent manual adjustments can be quickly learnt.

Response to Departure When house owners leave their house and their house are vacant, thermostats will maintain their current temperatures for 15 minutes. The time at which the user leaves is recorded as t_{leave}.

- If a house owner comes back within 15 minutes, the thermostat will continue functioning as if no one has left the house.
- If a house owner does not come back within 15 minutes, the thermostat will be turned off. It will be turned on again when the anticipated return time is reached.

Anticipation of Return Our system is able to anticipate a user's return based on the historical data of user's departure and return. Using data from the past two months, the system stores the probability distribution of the duration of a user's absence (x) given the time block the user leaves the house (t_{leave}). For example, if a user's period is 7 days, $\tau = 7 \times 96 = 672$(time blocks), and the user left her apartment on Sunday morning 9:00 AM $\left(t_{leave} = 6 \times 96 + \frac{60}{15} \times 9 = 612\right)$ for eight times in the past two months.

- Three times she went jogging for one hour(four time blocks), so $P(t_{abs} = 4 | t_{leave} = 612) = \frac{3}{8} = 0.375$.
- Four times she went to have a brunch with friends for two hours, so $P(t_{abs} = 8 | t_{leave} = 612) = \frac{4}{8} = 0.500$.
- She also went to her parents' house on one weekend and returned after six hours $P(t_{abs} = 24 | t_{leave} = 612) = \frac{1}{8} = 0.125$.

Using this information, we can generate a probability mass function of user's return against time. The use of a cumulative distribution function is less ideal, since the probability of return may not be monotonously increasing: for this particular user for example, $P(t_{return} = 612 + 4 \times 4 = 628 | t_{leave} = 612)$ may be less than $P(t_{return} = 612 + 4 \times 6 = 628 | t_{leave} = 612)$. Therefore, we decided to use a probability mass function with a slight change: the probability of return in all the times blocks after the latest possible return time is 1, as the user should have returned by that time. The adjusted probability mass function for the return time of this user is:

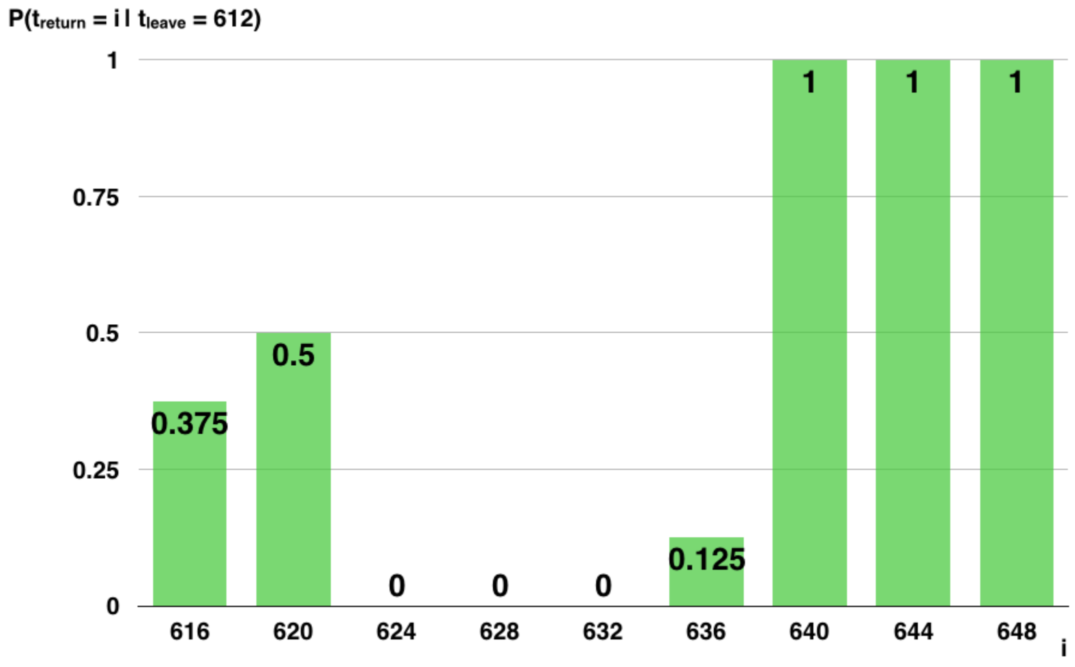

Figure 4.5. A diagram of the user's probability of return against time.

In predicting ideal temperatures for the day, we account for the user's return using a weighted utility function (see **Section 4.3.3, Utility Function**) with weights for satisfaction function as $P(t_{return} = i|t_{leave} = 612)$ where i is a specific time block. Intuitively, the user does not care about the temperature when he is not at home! The adjusted utility function is:

$$U(T_r^{[n]}(i)) = P(t_{return} = i|t_{leave} = 612)(\alpha S(T_p^{[n]}(i), T_r^{[n]}(i))) + \beta P(T_r^{[n]}(i)) \tag{36}$$

If the probability of return in a specific time block is small, the temperature in that time block will be closer to or even the same as the outdoor temperature as power matters much more than satisfaction, vice versa. Therefore, this adjustment is reasonable.

Irregular Schedule However, there can be cases in which the probability mass function for a particular t_{leave} spreads out across multiple time blocks, and has a very small peak value (maximum probability of return in a time block) or does not have a peak value at all. If it is the case, then we can say that the user's schedule for that period of time (from t_{leave} to the latest possible return time) is very irregular. Regarding this problem, if the aforementioned period of time is more than two hours and the peak value if less than 0.3, then the user's schedule will be considered as irregular. In this case, the system will advise the user to turn off "Anticipation of Return" function and, instead, ask the user to manually input his return time at least half an hour before his return. The system then prepares for his return by setting the room temperature to the ideal temperature.

4.4 Evaluation (1b)

To show the potential benefits and possible issues of our model, we compared our system with existing smart thermostats. We found two smart thermostats – Nest learning thermostat[10] and Ecobee[12]. We compared them in four aspects, including the ability of learning new trend, prediction of users' schedule, energy costs, and inclusion of additional factors.

4.4.1 Flexibility

Nest learning thermostat According to Nest's algorithm, it assumes that its users follow a five-weekdays-and-two-weekend-days schedule. How it learns from manual temperature changes is based with this assumption. For example, if a user changes the temperature for two weekdays in a row, temperatures in all weekdays afterwards will be changed. The table below describes its learning rules:

Manual Temperature Change	Future Automatic Changes
two weekdays in a row	all weekdays
two weekend days in a row	all weekend days
one weekday and one weekend day in a row	all days in a week

Table 4.3. The learning rules of Nest learning thermostat

Ecobee All Ecobee system's smart functions, such as Smart Home, Smart Away and Smart Recovery, are based on users' own setting of the schedule. In other words, it requires user to control the real time temperature using its mobile app, but it cannot really learn users' schedule and make automatic adjustments.

Our system Our system can learn schedules with periods of different duration, not just those with seven-day (five-weekdays-and-two-weekend-days) periods. It uses the detected period to adjust the temperatures.

Evaluation Compared to the fixed five-weekdays-and-two-weekend-days schedule of Nest thermostat, our thermostat can adapt to a variety of users' schedules and thus accommodate the diverse needs of users well. It is also more flexible about temperature changes. For example, it is impossible for Nest thermostat's users to change only the temperatures for all Fridays and Saturdays without changing the temperatures for all days afterwards. Our system, on the other hand, makes such changes possible. Furthermore, our system achieves this flexibility without compromising user friendliness: it can learn from users' past data and make automatic temperature changes, unlike Ecobee. Thus, flexibility is a distinct strength of our system.

4.4.2 Learning from Manual Temperature Changes

Nest learning thermostat As shown by Table 4.3, Nest thermostat learns based on a set of rigid rules. And once a change is made, information from previous days are all discarded and will not influence temperature changes in the future.

Ecobee Ecobee's temperature changes are all based on users' manual inputs. The system does not learn from them to make automatic temperature changes in the future.

Our system The learning process of our system, as elaborated in section **4.3**, does not follow strict rules. To predict the ideal temperature for the next day, it considers historical data from up to two months and assigns weights according to their expected predictive power. When a manual change is made, our system considers the change as an anomaly, and will not let it affect the prediction for the next day that much. However, if the same change is made in the next few days (typically 1-2 days), it will adapt to the change and predict future temperatures based on this knowledge. This change, unlike for nest learning thermostat, does not have to be a static value. It can be a general trend, either gradual or abrupt. For example, during transitions between seasons, users' preferred temperatures may follow a gradual decreasing or increasing trend. The user only needs to manually adjust temperatures at the beginning of the seasonal transition, and our algorithm will learn the rest.

Evaluation As Ecobee cannot learn from manual temperature changes, we will only be evaluating Nest and our product. Nest is superior to our learning algorithm in terms of transparency. Nest thermostat can be completely controlled because it is rule-based and the prediction process is completely transparent. Users can be sure how their manual inputs change the behaviours of the system. On the other hand, our algorithm is not based on simple rules, and although users are able to gauge the effect of their input to some accuracy, it will not completely accurate. However, this is by the fact that there are softer rules, such as "decrease temperature at this rate for two days, the temperatures will continue decreasing at this rate afterwards." Our system is superior to Nest in terms of adapting to changes. The change in Nest thermostat always has a lasting effect in the future, whereas for our system, a sudden change for one or two days may be considered as anomaly and will not have a lasting effect. This is advantageous as sometimes anomaly does occur (e.g. the house owner invites a friend with different temperature preference over to his house), and users do not need to make adjustments again if they are using our system. Nevertheless, it can be disadvantageous when a user may need to manually adjust temperatures for up to three days to actually make a permanent change. Our system is also superior in that it can predict trends and automatically adjust temperatures according to the trends.

4.4.3 Prediction of users' departure and arrival

Nest learning thermostat If everyone is away, the thermostat is set to a preset Eco Temperature. It would start pre-heating or pre-cooling when someone comes home or manually changes the temperature with the app.

Ecobee Smart Home and Smart Away: the adjustment is based on motion sensor to detect whether the user is at home. The thermostat switches from its Smart Home mode to Smart Away mode if no occupancts are detected for 2 hours.

Our system In our system, there are two modes for "Anticipation of Return" function. When the user's schedule is regular, we will advise him/her to turn on this mode. When this function is turned on, the system will record and accumulate data of the user's arrival and departure time. By analyzing data collected, it can calculate the probability of the user arriving or leaving the house in a particular 15-minute time block given the user's departure time. Instead of turning on and off the thermostat, we let it continue working but with different power. If the return probability is high, when setting the temperature, the satisfaction function (which approximates user's satisfaction) will be given a large weighting and the room temperature will be closer to user's preferred temperature. If the return probability is low, power consumption may be prioritized over satisfaction, so the air conditioner may be still turned on, but the temperature tends more towards the ambient temperature. When the function is turned off, our system asks users to manually input their return time so the air conditioner can be turned on in advance.

Evaluation While Nest learning thermostat gives users full control, if users set the eco-temperature to a temperature close to outdoor temperature and forget to reset the temperature remotely using the app, then they may feel very uncomfortable temperature when they return home. For a user who regularly leaves his apartment, such as a student, it is probable that he will occasionally forget to turn on the system in advance. If users set the eco-temperature close to his ideal temperature, then the system will incur much unnecessary cost and waste much energy when he is away. Thus, Nest's system design is not entirely satisfactory. For Ecobee, when users reach home, the temperature will also be the same as the outdoor temperature, so the user's comfort is compromised. Our system achieves a balance between comfort level and energy costs. Also, for users with relatively regular schedules, even if they forget to set their return time, they will still experience a comfortable temperature when they come back.

4.4.4 Energy costs

Nest learning thermostat It is noteworthy that Nest company had published a white paper on their energy savings analysis with field-tested results. Nest company surveyed utility bills of real households in 41 states in America before and after the installation of Nest learning thermostat, and concluded that, on average, Nest learning thermostat saved 10% to 12% on heating and 15% on cooling. The final sample size of 735 households for the gas usage analysis and 624 households for the electric analysis.

Ecobee According to the official website of ecobee, it is said that family in US that installed Ecobee thermostat saved up tp 23% on their heating and cooling. However, this figure of energy saving is based on a commonly used desired temperature of 22°C.

Compared to the data provided by the Nest company, Ecobee company only used a common set point of temperature for calculation, factors such as geographical locations, housing characteristics and so on could actually affect a lot on the calculated energy saving results. Hence, one problem with the energy saving data provided by Ecobee is that it may not accurately reflect expected savings in realistic circumstances.

Our system We also face the problem that we cannot conduct field tests but have only theoretical data. However, we found that the energy saving data provided by the two thermostat producers are only based on their investigation in America, which is mainly covered by subtropical and temperate zones. Due to the heterogeneous nature of a national sample, the data provided cannot apply to other countries or other climate zones such as frigid zone and tropical zone. In order to address this problem, we decide to investigate the energy saving of our system according to different seasons in different climatic zones respectively for specific temperature ranges. Theoretically, for temperate and subtropical zone in spring and autumn with temperature varied from 10 °C to 25 °C, our system could save 20% to 30%; for tropic zone with annual average temperature above 25 °C, our system could save 10% to 15% on cooling; whereas the saving can be from 25% to 35% in frigid zones. Hence, in energy saving aspect, our system is very eco-friendly and can be popularized to countries all over the world.

4.4.5 Other geographical conditions (humidity, allergen, and air purity level)

Other climate control systems are able to install additional components, including humidifiers and air purifiers, to control factors like humidity, allergens and air purity level. Meanwhile, our system only consists of a thermostat in the context of the question. Hence, the comparison between these currently-in-use climate control systems and our system cannot be effectively established. Although it is a limitation that we do not have additional components in our system, we are still able to consider these conditions in our model so as to improve our system's performance in real life situations.

5 Multi-zone Climate Control (Problem 2)

5.1 Introduction

We have designed a smart control system for a single person living in a small house with one thermostat. But in reality, many people live in a larger house with multiple cooling/heating zones. In order to address this situation, we modified our model in Problem 1 on account of users' different preferences and heat transfer between rooms.

5.1.1 Similarities and Differences between Multi-zone Systems and Single-zone System

Similarities The thermostat is able to perform all the functions that we have outlined below:

1. Predict the ideal temperature in the room throughout a day based on personal preference (historical temperature data) and power consumption.
2. Anticipate the users' arrival and adjust the temperature according to the users' preference before users reach their houses. (Noted that The first week's temperature need to be manually inputted by the user of the room).

Differences

1. In multi-zone systems, thermal interactions between zones should also be considered. Different zones, when set to different temperatures, influence the room temperatures in their adjacent zones. Thus, the power consumption and users' satisfaction in these zones are interdependent. It is important to consider how we can optimize global power consumption and users' satisfaction under this condition.

2. Nonetheless, quantifying thermal interactions is difficult in the multi-zone case. In single-zone system modelling, we are able to calculate one zone's thermal interaction with the outdoor environment based on various parameters such as the heat capacity of furniture and walls, aiming to achieve a high accuracy when calculating power consumption. On the contrary, in multi-zone systems, it is very hard to obtain the values of all these parameters as the number of parameters multiplies when there are many zones with each pair connected in a different way (walls connecting two rooms might be of different thickness and materials, some zones are connected by corridors or doors, etc.). We thus need to think of a way to approximate the collective influence of these parameters on inter-zone thermal interactions without asking users to manually measure them.

3. Many people may live in this system and spend time in the same cooling/heating zone. Different people have different preferences, so it will be good if our system can help them decide a temperature that can maximize the total comfort.

5.2 Learning Parameters

As mentioned in section **5.2**, single-zone systems are relatively homogeneous in terms of structure, building material and size, and the heat transfer is mainly between the outdoor environment and the room. The heat transfer in multi-zone systems are much more complex due to thermal interactions between zones. The constants of heat transfer also differ greatly. We need these values for predictions, but it would be troublesome for the user to measure them.

Interdependency Index Specifically, we focus on a parameter, k, *the interdependency index.* According to thermodynamics, the rate at which temperature in z_j changes is proportional to the rate at which heat is transferred into z_j. And the rate of heat transfer between two zones is proportional to the temperature difference between two zones, so temperature change in z_j due to zone z_k is directly proportional to the temperature differences between z_j and z_k. We denote the proportionality constant as k_{z_j,z_k}.

Suppose the house has a set of n zones (the outdoor environment is also counted as one zone) — $Z = \{z_1, z_2, ..., z_n\}$, and zone i ($z_i$) is adjacent to a set of m zone. For example, in the house below, $Z_5 = \{z_1, z_2, z_6, z_7\}$. We can model the house as a graph ($G = (Z, E)$), where Z is a set of nodes(zones) and E is a set of edges. If two zones, z_j, z_k are adjacent to each other, then there is an edge connecting them, and the weight on that edge is k_{z_j,z_k}. We can describe the graph using an adjacency matrix, and we want to obtain the weights of the edges – the non-zero values in this matrix.

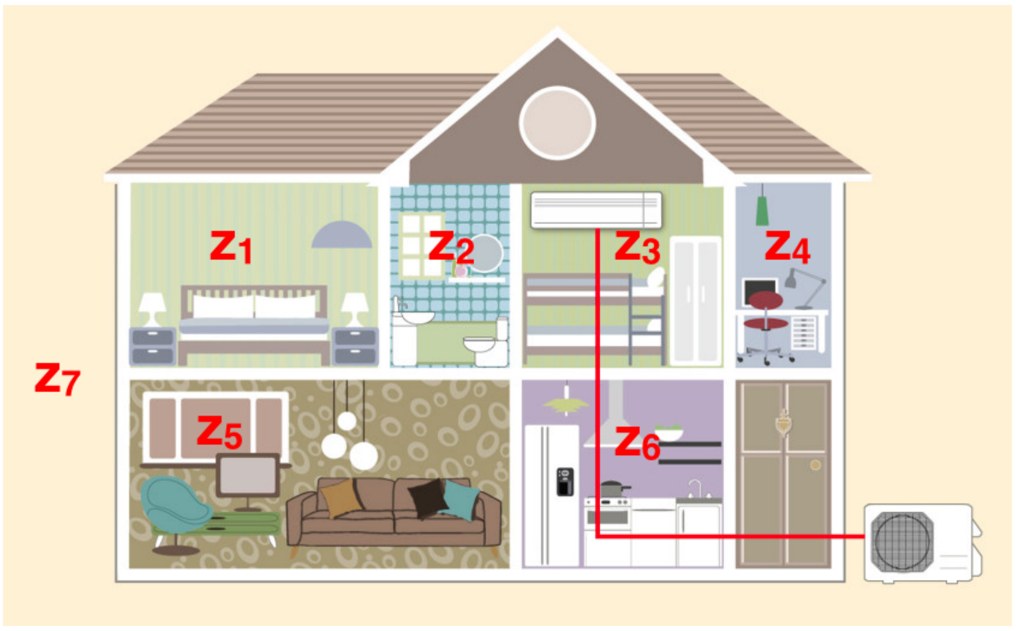

Figure 5.1. Zoning of a large house

There are $|E|$ unknown parameters. At any time, the temperature change in z_i, ΔT_{z_i}, determined by temperature in z_i (T_i) and those in its adjacent zones, can be expressed as:

$$\Delta T_{z_i} = \sum_{j=1}^{m_i} (T_{z_j} - T_i) k_{z_j,z_i} \tag{37}$$

Such equation is the linear equation that we needed to solve for the unknowns. We can obtain this equation by stopping the functioning of the air conditioner in z_i, control the room temperature in its adjacent zone so that it remains the same for a while, and record the temperature change in z_i and the temperatures in its adjacent zones. The user's manual for our product will ask our users to turn on this "calibration" mode when the system is run for the first time. In this mode, the system will carry out these tests and gather enough equations to solve for the interdependency indices. After that, the mode can be turned off and the parameters learnt can help the system to accurately control the temperatures in different zones simultaneously.

5.3 Thermal Interactions between Zones

After the interdependent indices have been measured, we organize them into a square matrix K, with elements

$$\mathbf{K} = \begin{pmatrix} k_{z_1,z_1} & k_{z_1,z_2} & \cdots & k_{z_1,z_a} \\ k_{z_2,z_1} & k_{z_2,z_2} & \cdots & k_{z_2,z_a} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \tag{38}$$

The diagonal elements should be chosen so that the sum of each row is 1. This is because at the next time block the temperature of room i is given by

$$T_i(t+\Delta t) = T_i(t) + \frac{\Delta t}{C_i} \sum_{i \neq j} k_{ij}(T_j - T_i) = T_i(t) + \sum_{i \neq j} k_{z_i,z_j}(T_j - T_i) \tag{39}$$

If we choose k_{z_i,z_i} according to equation

$$k_{z_i,z_i} = 1 - \sum_{i \neq j} k_{z_i,z_j} \tag{40}$$

so that the row sum is just 1, we can conveniently write down the formula for the next temperature as

$$T_i(t + \Delta t) = \sum_{j=1}^{n} k_{z_i,z_j} T_j \tag{41}$$

which we easily recognize as a matrix multiplication. The last row in matrix $\mathbf{K}$ is $(0, 0, \cdots, 1)$ because the ambient temperature is, by assumption, not affected by the temperatures inside the house. Of course, in its current form this equation does not permit the ambient temperature to change. However, this is only a minor issue in view of the discretization process we are taking. At each stage, we are only interested in how much energy must be spent to change the temperatures to desired ones. Since change in ambient temperature is not determined by the owner of the house, no energy cost is incurred in changing it anyways. And since we have discretization, we can manually change the value of ambient temperature when performing each step in our computations without actually affecting the outcomes.

As mentioned before, if we write the temperatures inside each of the zones as a column vector,

$$\mathbf{T} = \begin{pmatrix} T_1 \\ T_2 \\ \vdots \\ T_a \end{pmatrix} \tag{42}$$

If the air conditioner is not turned on, the temperature vector at the next time block should be just

$$\mathbf{T}_{\text{natural}} = \mathbf{KT} \tag{43}$$

The difference between this natural temperature and the desired temperature is the effort that has to be done by the air conditioners. We can adjust the elements according to formula (28) and sum the elements together. That is the net work that has to be supplied by the air conditioners and is the direct generalization of the formula for the single zone case.

5.4 Conflicting Needs of Users in the same House

Private Place and Public Places There are two types of zones in a house with multiple cooling/heating zones. We define

- a place with only one specific person staying in it most of the time to be a private place, e.g. a single room.
- a place with more than one person sharing it is defined as a public place, e.g. living room, bathroom, a bedroom for two people and etc.

As what we have mentioned above, there are some similarities between a multi-zone system and a single-zone one, hence we can still relate the two situations in some ways.

We consider each heating or cooling zone as a small apartment. In this way, private places can be considered the same as the situation we have addressed in problem 1 as no conflict on temperature setting will occur in a private place.

For public spaces and the interactions between different zones, we can address these situations by adding on other considerations so as to maximize everyone's comfort.

Satisfaction Function for Many Residents Facing the problem of how to set the temperature when several people with different preferences are in the same cooling/heating zone, we use the following method: Since one person's satisfaction value at one specific temperature is representative of his likelihood of agreeing on this temperature with others, we try to maximize the product of different people's satisfaction value so as to get the optimal temperature - the one most likely to be accepted by everyone.

One example illustrates this clearly. Suppose that there are two people, one prefers 24°C while the other prefers 26°C. They have the same standard deviation in their preference. If we neglect power consumption for now, one could argue that, somewhat obviously, 25°C is the best choice. If we were to sum up the two satisfaction functions, we will find that, at least for smaller standard deviations, there are two local maxima at 24 and 26°C, while 25°C is a local minimum. On the other hand, if we use the product, there is a maximum at 25°C, whereas the peaks at 24 and 26°C are exponentially suppressed by each other.

Since the original peaks are exponentially suppressed, very often the new combined satisfaction function will be approximately a normal distribution. It will be skewed if different people have different standard deviations in their preference, but this means that even without knowing the satisfaction functions of everyone in the room, we can still have the system learn from the collective behavior of the residents, just like how we would learn from the behavior of a single person in a small room. Of course, the assumption here is that the residents will be considerate in public settings instead of being self-centered, but arguably this assumption is not too far-fetched.

6 Robustness and Sensitivity Analysis

6.1 Period Identification

Our algorithm detects period underlying a user's schedule using auto-correlation analysis. The user's period will be the smallest time lag (τ) with an R value ($R(\tau)$) (autocorrelation coefficient) larger than 0.7. $R(\tau)$ is a measure of the correlation between two periods of data given that the length of the period is τ. The maximum $R(\tau)$ also reflects the regularity of user's schedule. One meaningful question to ask, then, is how does our prediction accuracy for ideal temperature and anticipation of return time change as we accept a period with a τ value different from 0.7, and if the setting 0.7 as the threshold is reasonable.

However, in reality, ”irregularity in schedule” usually points to the fact that the user leaves and returns home at irregular timings, but not that the user change temperatures irregularly. Most likely, the user's preferred temperature throughout a particular day in a period remains the same in recent two months unless affected by long-term changes or short term anomaly (discussed in the next section). Therefore, we will only discuss how the prediction accuracy for return time change as we accept periods with different τ values.

We created some datasets and set their periods so that they have different $R(\tau)$ values. Our algorithm was then ran on each dataset. For each dataset, the algorithm generated a probability mass function (pmf) of return probability for each time block in a period (for details, please refer to section **4.3.5**). It then calculates the average peak value of all the pmfs for a dataset.As we cannot control $R(\tau)$ precisely, we generated a scatter plot based on the test results. The x-axis is the specific $R(\tau)$ values, and the y-axis is the average peak values of pmfs of all the time blocks from a user's schedule with a particular $R(\tau)$ value.

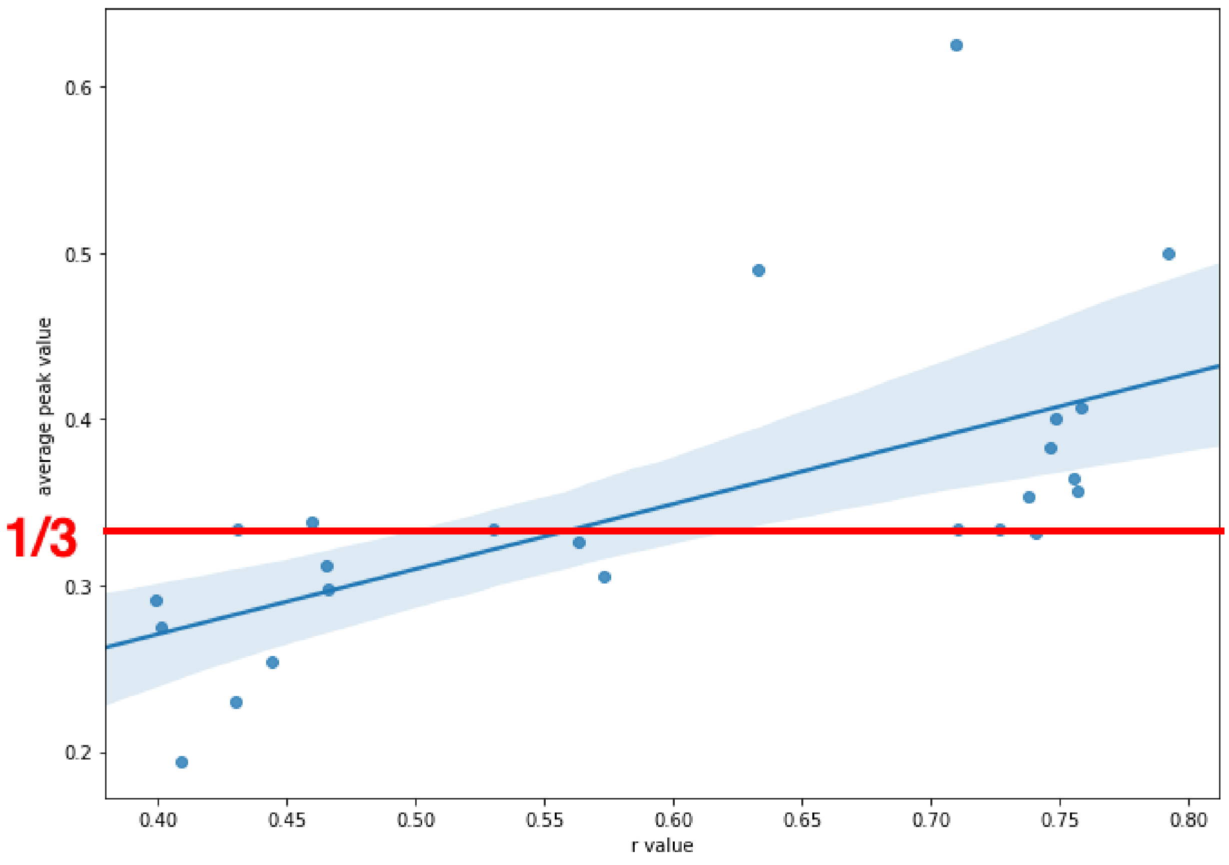

Figure 6.1. A graph of average peak value against r value

We can observe that generally as $R(\tau)$ increases, the average peak value increases. Furthermore, when $R(\tau)$ is less than 0.6, the peak values are mostly below 1/3, meaning that the maximum probability of return in a time block is less than $\frac{1}{3}$. Considering the design of our system, this means that when $R(\tau)$ is small (below 0.6), the weight given to the satisfaction function (the probability of return in a time block) will be small, and consequently the room temperature when the user actually returns will be not so different from outdoor temperature, which makes the system's "Return Anticipation" function useless. This is why when $R(\tau)$ is below 0.7, though we automatically set the period to 7 days for temperature prediction, we will advise user to turn off "Return Anticipation" function and instead manually input their return time using our mobile app prior to their return. We set the threshold to 0.7 because we do not have enough data for $0.6 < R(\tau) < 0.7$ to be certain that the peak value corresponding to $R(\tau)$ in this range is large enough, as we faced difficulty when trying to generate temperature data with $R(\tau)$ value in this range. In conclusion, 0.7 is a reasonable threshold.

6.2 Personal Preference - Prediction based on Historical Data

Our algorithm predicts the ideal temperature throughout a new day based on power consumption calculation and past data, which can reveal our users' personal preferences of temperature and their schedules. However, the historical data may not be the same throughout, for as seasons and their schedules change, their preferred temperatures and periods in their schedules may change, gradually or abruptly. Also, there may be anomalies in past data due to short-term weather change or other reasons such as the visit of elderly to the house. In this section, we will test the robustness of our algorithm by testing the accuracy of predictions under different simulated situations.

Gradual Temperature Shifting - Summer to Fall The temperatures at which the system was set generally decreases from 28°C to 18°C with some irregularities in between (random slight temperature increase), because as fall comes the user generally wears more indoor (shorts to pants, etc.).The performance of our algorithm on this dataset is described in the table below. n is the number of days out of 21 days we predicted that have an absolute error larger than 1.5°C.

Mean Absolute Error/°C	Max Absolute Error/°C	Mean Percentage Error/%	n
0.28	0.92	1.56	0

Table 6.1. Mean absolute error, max absolute error and mean percentage error of our algorithm.

We can see that our algorithm deals with the gradual change in temperature well, as the prediction follows the trend of actual temperature.

Tapering Off - Back to Stable Temperatures These two datasets simulate the tapering of effect at the end of temperature shifting. The temperature shifting in the first dataset(**Gradual**) is more gradual compared to that in second(**Abrupt**), as it spans two weeks and the latter spans six days. The additional metrics, "Reaction Time", is the number of days taken for the system to realize that temperature change is tapering off (to stability).

	Gradual	**Abrupt**
Mean Absolute Error/°C	0.57	0.67
Max Absolute Error/°C	1.41	1.55
Mean Percentage Error/%	2.51	2.75
n	0	0
Reaction Time/days	1	1

Table 6.2. Prediction accuracy of our algorithm on two data sets (**Gradual** and **Abrupt**)

We can observe from the table that the system is able to learn quickly and accurately when the temperature change ends. Its short reaction time indicates that the user only needs to input data manually for one day and the algorithm will learn that the shifting has ended. It will consequently assign weights to more recent data and ignore the shifting effect from more previous days to accommodate the change quickly.

Abnormalities These three data sets simulate abnormalities in temperatures, possibly due to events such as a sudden cold weather, the visit of elderly, sickness of some users, etc. The abnormality in each data sets spans time periods of different lengths (**2 days**, **4 days** and **8 days**).

	2 days	**4 days**	**8 days**
Mean Absolute Error/°C	0.99	0.91	1.01
Max Absolute Error/°C	4.92	3.22	3.26
Mean Percentage Error/%	4.16	3.84	4.13
n	2	2	2
Reaction Time/days	1	1	1

Table 6.3. Prediction accuracy of our algorithm on three data sets with abnormalities of different length (**2 days**, **4 days** and **8 days**)

It can be observed from the table that all the measurements for errors increase. This is because at the "edge" of abnormalities - when transiting from abnormal period to normal period, the errors are large for one/several day(s) (as indicated by Reaction Time). However, we can see that the Reaction Time is still short, indicating that the user only needs to manually adjust the temperatures for one day and the system will learn that abnormal period has passed and it will assign larger weights to periods before and after the abnormal period to accommodate this change.

6.3 Utility Function

By adjusting α and β, the weightings we put on satisfaction and power consumption respectively, we can adjust the system's behavior. If we make α larger, it would mean that the system focuses on keeping the users happy and the reduction in energy usage is not very large. On the other hand, if we make β larger, the system would care less about the comfort of the user and focuses more on saving energy. Now, it should be mentioned that the c parameter in MCTS that dictates how much exploitation is done over exploration should be changed according to how we change the parameters. In general, one would set c to be at least a few million, so that all options have a decent chance of being selected.

With 10,000 roll-outs at each state, the test results are shown in table 6.3. With optimization and better heuristics, we can achieve even greater reduction, but the main focus should be that indeed, control of the system can be achieved by simply tweaking the two parameters and making necessary changes to other corresponding parameters, such as c.

Unfortunately, changing the parameters by less than a factor of 10 has less of an impact on the system behavior due to the random nature of the algorithm we are employing. If the change in the parameter is only by a factor of two, random fluctuations in the algorithm's roll-out process may easily wash out effects of the change. This is something to consider when choosing appropriate parameters for a particular geographical location and climate.

α, β	5000, -1	5000, -10
Tropical weather	$9.1 \pm 0.7\%$	$13.8 \pm 0.6\%$
Temperate weather	$15.6 \pm 0.6\%$	$20.3 \pm 0.6\%$
Frigid weather	$29.3 \pm 0.8\%$	$38.2 \pm 0.7\%$

Table 6.4. Energy Reduction under Various Conditions

7 Strengths and Weaknesses

7.1 Strengths

1 Our system does not make assumptions about users' schedules, so it can adapt to any type of schedules. Unlike other smart home climate control systems, our systems do not assume the that our users all follow a five-weekday-two-weekend-day (5-2) schedule, which increases the accuracy of both prediction of ideal temperature changes and estimation of duration of absence, especially for users with schedules different from 5-2.

2 Our system learns a change in preferred temperature quickly due to the flexibility of algorithm (the assigned weights change). It can tell when an anomaly or a gradual temperature shift starts and ends with a time lag of one to two days. This means that the user only needs to manually adjust the temperatures for one or two consecutive days to set the ideal temperatures afterwards to the right track (i.e. as he/she desired).

3 Our system takes power consumption into consideration, and it can reduce electricity bill significantly without compromising users' comfort.

4 Our system is very convenient to use, especially for users with a relatively regular schedule, provided that few abnormal events occur. They need not to manually input anything after the first period passes. They need not even to remotely set their return time as the system will learn that by itself.

5 Inter-zone thermal interactions and differences between inter-zone connections can increase the inaccuracy of temperature control, causing the actual room temperature to differ significantly from the temperature set by the system. The multi-zone climate control system we designed takes these two factors into consideration and thus can improve the accuracy of temperature control greatly.

6 Our system provides a method for balancing conflicting needs of multiple people in a reasonable way (see section **5.4**). This function may reduce the number of potential disputes and thus help fostering harmony among family members/house owners.

7.2 Weaknesses

1 Our system cannot identify irregular periods. For example, if a person (e.g. a doctor) takes 10 days off in a month, then his period is likely to be a month. But months differ in lengths. Our system may assume the period to be 30 days, which will cause some inaccuracy in prediction. Alternatively, it may identify the person's schedule as irregular and require more manual inputs, which will cause some inconvenience to the user.

2 Our algorithm can be computationally expensive due to the large number of iterations Monte Carlo Tree Search requires to obtain optimal results. To predict ideal temperatures for one day, it can take up to five minutes.

3 The ambient temperature data we used to calculate ideal temperature for the next day is obtained through local weather forecasting, which may be inaccurate.

4 The calculation of power consumption for multi-zone systems is a rough approximation, as the temperature change in a zone is not only due to thermal interactions between zones and between outdoor environment and the zone, but also a myriad of other factors, such as what furniture is added to the room. The single-room system takes some of these factors into consideration and is thus relatively more accurate.

5 We did not take thermal radiations, especially solar radiation which can affect room temperature significantly, into consideration when calculating power consumption. This is because, as explained in section **4.3.2**, they are too complex to be modeled accurately.

References

[1] Understanding human comfort… more than just temperature and humidity. *Indoor Air Quality in Northwest Schools*, Spring 2010.

[2] Miimu Airaksinen, Pekka Tuomaala, and Riikka Holopainen. Modeling human thermal comfort. *Proceedings of Clima*, 2007.

[3] Giovanna Aronne. Effects of relative humidity and temperature stress on pollen viability of cistus incanus and myrtus communis. *Grana*, pages 364–367, 1999.

[4] Simon A. Carn. Atmospheric transmission.

[5] William de Ferranti. Piecewise smooth extremals.

[6] DuPont Suva refrigeriants. *Thermodynamic Properties of HFC-134a*.

[7] HyperPhysics. Relative humidity.

[8] Preeti Kannan. Sudden change from hot to cold can harm health, Aug 2012.

[9] Jackson Kung'u. Factors that affect the growth of moulds.

[10] Nest Labs. Nest white paper.

[11] Inc. Mikron Instruments Company. Table of emissivity of various surfaces.

[12] Andrew Newton. Feature friday: Follow me, and smart home/away, 2015.

[13] Jennifer L. Nguyen, Joel Schwartz, and Douglas W. Dockery. The relationship between indoor and outdoor temperature, apparent temperature, relative humidity, and absolute humidity. *Indoor Air*, 24(1):103–112, Feb 2014.

[14] Daniel V. Schroeder. *An Introduction to Thermal Physics*. Addison Wesley, 1 edition, Aug 1999.

[15] Zhou Tingyu. Experts: 6 tips on getting rid of mite-induced allergies, Jun 2009.

[16] Engineering Toolbox. Indoor comfort temperature versus outdoor temperature.

[17] Engineering Toolbox. Air - specific heat at constant pressure and varying temperature, 2018.

[18] Designing Buildings Wiki. Specific heat capacity, Nov 2016.

[19] Wikipedia. Autocorrelation.

[20] Wikipedia. Thermal comfort.

Appendices

Periodicity Detection Code

```
import numpy as np
import pandas as pd

def find_period(df,mini=1,maxi=31):
    '''
    input: pd.Series object containing historical temperature data
    output: the most likely period
    '''
    r=[]
    for i in range(1,32):
        old=df.copy(deep=True)
        new=df.copy(deep=True)
        old=old.drop(old.index[list(range(df.shape[0]-96*i,df.shape[0]))])
        new=new.drop(new.index[list(range(0,96*i))])
        c=pd.concat((old,new),axis=1)
        r.append(pd.Series(df).autocorr(lag=i*96))
    period=7
    for i in range(31):
        if r[i]>0.7:
            period=i+1
            break
    return period
```

Monte-Carlo Tree Search Code

```
#!/usr/bin/ruby

class TempComputation
  # COP refers to COP of cooling.
  @@c_wall = 3000000
  @@c_room = 150000
  @@conduction_rate = 800
  @@interval = 15 * 60.0
  @@cop = 4.0
  @@alpha = 500.0
  @@beta = -2.0

  def self.get_dv_temp(temp, next_temp)
    (next_temp - temp) / @@interval
  end

  def self.power(temp, temp_wall, dv_temp)
    qr = @@conduction_rate * (temp_wall - temp) - @@c_room * dv_temp
    if qr >= 0 then
      return qr * @@interval / @@cop
    else
      return -qr * @@interval / (@@cop + 1)
    end
  end

  def self.utility(temp_state)
    @@alpha * temp_state.user.satisfaction(temp_state.temp, temp_state.time)
  end

  def self.next_temp_wall(temp_room, temp_wall, temp_amb)
    temp_wall + (@@interval * @@conduction_rate * (temp_room + temp_amb - 2.0
  end
end

class UserState
  def initialize(left_time, pdf_return, temp_fav, stddev)
    @left_time = left_time
    @pdf_return = pdf_return
    @cdf_return = compute_cdf
    @temp_fav = temp_fav
    @stddev = stddev
  end

  def at_home?(time)
    if time < @left_time then
      return 1.0
    else
      return @cdf_return[time - @left_time]
    end
  end

  def satisfaction(temp, time)
    Math.exp(-(temp - @temp_fav[time])**2 / (2.0 * @stddev**2)) * at_home?(ti
  end

  private
  def compute_cdf
    cdf = [ @pdf_return[0] ]
    (1...@pdf_return.size).each do |i|
      cdf.push(cdf[-1] + @pdf_return[i])
    end
    return cdf
  end
end

class TempState
  attr_accessor :temp, :user, :time, :temp_wall, :dv_temp

  @@ambient = [11, 11, 11, 11, 11, 11, 11, 11,
               11, 11, 11, 11, 11, 11, 11, 11,
               11, 11, 11, 11, 12, 12, 12, 12,
               12, 12, 12, 12, 13, 13, 13, 13,
               14, 14, 15, 15, 15, 15, 16, 16,
               17, 17, 18, 18, 19, 19, 19, 19,
               20, 20, 21, 21, 22, 22, 22, 22,
               21, 21, 21, 21, 20, 20, 20, 20,
               19, 19, 19, 19, 18, 18, 17, 17,
               16, 16, 16, 16, 15, 15, 15, 15,
               14, 14, 14, 14, 13, 13, 13, 13,
               12, 12, 12, 12, 11, 11, 11, 11]

  def initialize(temp, time, user, temp_wall, dv_temp)
    @temp = temp
    @time = time
    @user = user
    @temp_wall = temp_wall
    @dv_temp = dv_temp
  end

  def legal_moves
    (-16..16).map { |i| @temp + i * 0.125 }.reject { |temp| temp < 16 or temp
  end

  def last_block?
    @time == 95
  end

  def next_state(temp)
    TempState.new(temp, time + 1, @user, TempComputation.next_temp_wall(temp,
  end
end

class MonteCarloTreeNode
  attr_accessor :q, :n, :state, :parent, :children

  def initialize(state, parent)
    @state = state
    @untried = state.legal_moves
    @children = []
    @parent = parent
    # results
    @q = 0
    # number of visits
    @n = 0
  end

  def rollout
    current_rollout = @state
    net_util = TempComputation.utility(current_rollout)
    until current_rollout.last_block? do
      current_rollout = current_rollout.next_state(rollout_policy(current_rol
      net_util += TempComputation.utility(current_rollout)
    end
    return net_util
  end

  def expand
    child = MonteCarloTreeNode.new(@state.next_state(@untried.pop), self)
    @children.push(child)
    return child
  end

  def fully_expanded?
    @untried.empty?
  end

  def back_propagate(net_util)
    @n += 1
    @q += net_util
    if @parent then
      @parent.back_propagate(net_util)
    end
  end

  def is_terminal_node?
    return @state.last_block?
  end

  def best_child(c_param = 50000000)
```

```
    @children.max_by { |c| (c.q.to_f / c.n.to_f) + c_param * Math.sqrt(Math.l
  end

  private
  def rollout_policy(moves)
    return moves.sample
  end
end

class MonteCarloTreeSearch
  attr_accessor :root
  @@simulations = 12000

  def initialize(root)
    @root = root
  end

  def best_action
    @@simulations.times do |i|
      leaf = tree_policy
      leaf.back_propagate(leaf.rollout)
    end
    @root.best_child(c_param = 0.0)
  end

  def output_children
    @root.output_children
  end

  private
  def tree_policy
    current_node = @root
    until current_node.is_terminal_node? do
      unless current_node.fully_expanded?
        return current_node.expand
      else
        current_node = current_node.best_child
      end
    end
    return current_node
  end
end

temp_fav = [19] * 96
user = UserState.new(100, [], temp_fav, 2.5)
search = MonteCarloTreeSearch.new(MonteCarloTreeNode.new(TempState.new(17, 0,

ncost = 0
nsati = 0
nutil = 0
```

```
i = 1

until search.root.state.last_block?
  action = search.best_action
  ncost += TempComputation.power(search.root.state.temp, search.root.state.te
  nsati += user.satisfaction(search.root.state.temp, search.root.state.time)
  nutil += TempComputation.utility(search.root.state)
  puts "Predicted temperature for time block #{i}: #{action.state.temp}"
  puts "Predicted wall temperature: #{action.state.temp_wall}"
  puts "Cumulative cost: #{ncost}"
  puts "Cumulative satisfaction: #{nsati}"
  puts "Cumulative utility: #{nutil}"
  puts "\n\n"
  action.parent.children.each do |c|
    if c != action then
      c = nil
    end
  end
  action.parent = nil
  search = MonteCarloTreeSearch.new(action)
  i += 1
end
```

Ideal Temperature Prediction (Personal Preference) Code

```
import numpy as np
import pandas as pd

def recent(r2,r1):
    return (r2-r1).sum()/r1.shape[0]

def weight(ts,history=4):
    val=ts[history]
    ts=ts[:history]
    diff=abs(val-ts)
    m=max(diff)
    w=(m+0.001)-diff
    for i in range(history-1):
        w[i]=w[i+1]
    s=sum(w)
    w=w/s
    loss=(min(diff))/(sum(ts))
    if val<min(ts):
        w=w-loss
    if val>max(ts):
        w=w+loss
    return w

def batch_predict(ts,position,period=7,reverse=False,history=4):
    if position!=period:
        y=ts[-96*(period-position+1):-96*(period-position)]
    else:
        y=ts[-96:]
    print(ts)
    x=np.asarray(ts[-96*(history+2)*period:]).reshape((history+2,period*96))
    if position!=period:
        x=x[:-1,-96*(period-position+1):-96*(period-position)]
    else:
        x=x[:-1,-96:] #get the last day's historical data
    w=np.zeros((history,1))
    for i in range(96):
        w=np.concatenate((w,weight(x[:,i],history=history).reshape(history,1)
    w=w[:,1:]
    if position!=period:
        s1=-96*(period-position+1)-96*period
        e1=-96*(period-position+1)
        s2=-96*(period-position+1)-2*96*period
        e2=-96*(period-position+1)-96*period
        result=np.multiply(w,x[1:,:]).sum(axis=0)+recent(ts[s1:e1],ts[s2:e2])
    else:
        result=np.multiply(w,x[1:,:]).sum(axis=0)+recent(ts[-96*2:-96],ts[-9
    r=list(result)
    return (result,y,r)
```

```
def batch(ts,period=7,history=4,n=3):
    mean=[] #mean actual temperature
    mean_pred=[] #mean predicted temperature
    mean_abs_error=[] #mean absolute error
    max_abs_error=[] #max absolute error
    count=0
    for j in range(n):
        for i in range(1,period+1):
            (result,y,_)=batch_predict(ts,i,period=period,history=history)
            m=result.sum()/96
            ae=np.absolute(result-np.asarray(y)).sum()/96
            me=max(list(np.absolute(result-np.asarray(y))))
            if ae>1.5: count+=1
            mean.append(y.sum()/96)
            mean_pred.append(m)
            mean_abs_error.append(ae)
            max_abs_error.append(me)
        ts=ts[:-96*period]
    return(mean,mean_pred,mean_abs_error,max_abs_error,count)
```

Probability Mass Function Generation Code

```
import numpy as np
import pandas as pd

def pmf(df,n_days,period):
    '''
    df: pd.Series object
    period: period in days
    '''
    duration=[]
    mi=[]
    q=n_days//period
    r=n_days%period
    df=df[r*96:]
    df=np.asarray(df).reshape((q,period*96))
    for i in range(96*period):
        dur_dic={} #frequency dictionary
        for j in range(q):
            if df[j,i-1]!=1 and df[j,i]==1:
                d=dur(df,(j,i),period)
                if d in dur_dic.keys():
                    dur_dic[d]+=1
                else:
                    dur_dic[d]=1
        if len(dur_dic)!=0 and sum(dur_dic.values())>2:
            m=max(dur_dic.values())/sum(dur_dic.values())
            spread=max(dur_dic.keys())-min(dur_dic.keys())
            duration.append((dur_dic,m,spread))
            mi.append(m)
    return (duration,mi)

def dur(df,pos,period):
    '''
    df: 2-D matrix, each row is a period of temperature data
    pos:(period number, time block numer), 0-indexed
    '''
    count=0
    for i in range(pos[1]+1,period*96):
        if df[pos[0],i]==1:
            count+=1
    return count
```

Temperature Data from Different Groups of People As the entire data set is extremely large, we are only able to include part of the data set in this paper. Please refer to next page for the data set.

A student's schedule on school days

Time block	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
1	25	26	26	26	26	26	25	25	26	26
2	25	26	26	26	26	26	25	25	26	26
3	26	26	26	26	26	26	25	25	26	26
4	26	26	26	26	26	26	26	25	26	26
5	26	26	26	26	26	26	26	25	26	26
6	26	26	26	26	26	26	26	26	26	26
7	26	26	26	26	26	26	26	26	26	26
8	26	26	26	26	26	26	26	26	26	26
9	26	26	26	26	26	26	26	26	26	26
10	26	26	26	26	26	26	26	26	26	26
11	26	26	26	26	26	26	26	26	26	26
12	26	26	26	26	26	26	26	26	26	26
13	26	26	26	26	26	26	26	26	26	26
14	26	26	26	26	26	26	26	26	26	26
15	26	26	26	26	26	26	26	26	26	26
16	26	26	26	26	26	26	26	26	26	26
17	26	26	26	26	26	26	26	26	26	26
18	26	26	26	26	26	26	26	26	26	26
19	26	26	26	26	26	26	26	26	26	26
20	26	26	26	26	26	26	26	26	26	26
21	26	26	26	26	26	26	26	26	26	26
22	26	26	26	26	26	26	26	26	26	26
23	26	26	26	26	26	26	26	26	26	26
24	26	26	26	26	26	26	26	26	26	26
25	26	26	26	26	26	26	26	26	26	26
26	26	26	26	26	26	26	26	26	26	26
27	26	26	26	26	26	26	26	26	26	26
28	26	26	26	26	26	26	26	26	26	26
29	26	26	26	26	26	26	26	26	26	26
30	26	26	26	26	26	26	26	26	26	26
31	26	NA	NA	NA	NA	NA	26	26	NA	NA
32	26	NA	NA	NA	NA	NA	26	26	NA	NA
33	26	NA	NA	NA	NA	NA	26	26	NA	NA
34	26	NA	NA	NA	NA	NA	26	26	NA	NA
35	26	NA	NA	NA	NA	NA	26	26	NA	NA
36	26	NA	NA	NA	NA	NA	26	26	NA	NA
37	26	NA	NA	NA	NA	NA	26	26	NA	NA
38	26	NA	NA	NA	NA	NA	26	26	NA	NA
39	26	NA	NA	NA	NA	NA	26	26	NA	NA
40	25	NA	NA	NA	NA	NA	26	25	NA	NA
41	25	NA	NA	NA	NA	NA	25	25	NA	NA
42	24	NA	NA	NA	NA	NA	25	25	NA	NA
43	NA	NA	NA	NA	NA	NA	26	25	NA	NA
44	NA	NA	NA	NA	NA	NA	26	25	NA	NA
45	NA	NA	NA	NA	NA	NA	26	25	NA	NA
46	NA	NA	NA	NA	NA	NA	26	24	NA	NA

47	NA	NA	NA	NA	NA	NA	26	24	NA	NA
48	NA	NA	NA	NA	NA	NA	26	24	NA	NA
49	NA	NA	NA	NA	NA	NA	26	24	NA	NA
50	NA	NA	NA	NA	NA	NA	26	24	NA	NA
51	NA	NA	NA	NA	NA	NA	26	24	NA	NA
52	NA	NA	NA	NA	NA	NA	26	24	NA	NA
53	NA	NA	NA	NA	NA	NA	24	24	NA	NA
54	24	NA	NA	NA	NA	NA	24	24	NA	NA
55	24	NA	NA	NA	NA	NA	24	24	NA	NA
56	24	NA	NA	NA	NA	NA	25	24	NA	NA
57	24	NA	NA	NA	NA	NA	25	24	NA	NA
58	24	NA	NA	NA	NA	NA	25	24	NA	NA
59	24	NA	NA	NA	NA	NA	25	NA	NA	NA
60	25	NA	NA	NA	NA	NA	25	NA	NA	NA
61	25	24	NA	NA	NA	NA	26	NA	NA	NA
62	25	24	NA	NA	NA	NA	26	NA	NA	NA
63	25	24	NA	NA	NA	NA	26	NA	NA	NA
64	23	25	NA	23	NA	NA	26	NA	NA	NA
65	23	25	NA	23	NA	NA	26	NA	NA	NA
66	23	25	NA	23	24	NA	26	NA	25	NA
67	23	25	NA	23	24	NA	26	NA	25	NA
68	23	25	NA	23	24	NA	24	NA	25	NA
69	25	25	NA	23	24	NA	24	NA	25	NA
70	25	25	24	23	24	NA	24	25	25	24
71	25	25	24	25	24	NA	24	25	25	24
72	25	25	24	25	24	22	23	25	25	24
73	25	25	24	25	24	22	23	25	25	24
74	25	NA	24	25	24	22	25	25	25	24
75	25	NA	24	25	NA	22	25	25	25	24
76	24	NA	24	25	NA	24	25	24	25	24
77	24	NA	23	25	NA	24	25	24	25	23
78	24	NA	23	24	NA	24	25	24	25	23
79	24	NA	23	24	NA	24	25	24	25	23
80	25	24	23	24	24	25	24	25	24	23
81	25	24	23	25	24	25	24	25	24	23
82	25	25	25	25	25	25	24	25	25	25
83	25	25	25	25	25	25	25	25	25	25
84	25	25	25	25	25	25	25	25	25	25
85	25	25	25	25	25	25	25	25	25	25
86	25	25	25	25	25	25	25	25	25	25
87	25	25	25	25	25	25	25	25	25	25
88	25	25	25	25	25	25	25	25	25	25
89	25	25	25	25	25	25	25	25	25	25
90	25	25	25	25	25	25	25	25	25	25
91	25	25	25	25	25	25	25	25	25	25
92	25	26	25	25	25	25	25	25	25	26
93	25	26	25	25	25	25	25	25	25	26
94	26	26	25	25	26	25	25	26	26	26
95	26	26	25	25	26	25	25	26	26	26
96	26	26	25	25	26	25	25	26	26	26

Time block	Day 11	Day 12	Day 13	Day 14	Day 15	Day 16	Day 17	Day 18	Day 19	Day 20
1	26	26	26	25	25	26	26	25	25	26
2	26	26	26	25	25	26	26	25	25	26
3	26	26	26	26	25	26	26	25	26	26
4	26	26	26	26	25	26	26	25	26	26
5	26	26	26	26	25	26	26	25	26	26
6	26	26	26	26	26	26	26	26	26	26
7	26	26	26	26	26	26	26	26	26	26
8	26	26	26	26	26	26	26	26	26	26
9	26	26	26	26	26	26	26	26	26	26
10	26	26	26	26	26	26	26	26	26	26
11	26	26	26	26	26	26	26	26	26	26
12	26	26	26	26	26	26	26	26	26	26
13	26	26	26	26	26	26	26	26	26	26
14	26	26	26	26	26	26	26	26	26	26
15	26	26	26	26	26	26	26	26	26	26
16	26	26	26	26	26	26	26	26	26	26
17	26	26	26	26	26	26	26	26	26	26
18	26	26	26	26	26	26	26	26	26	26
19	26	26	26	26	26	26	26	26	26	26
20	26	26	26	26	26	26	26	26	26	26
21	26	26	26	26	26	26	26	26	26	26
22	26	26	26	26	26	26	26	26	26	26
23	26	26	26	26	26	26	26	26	26	26
24	26	26	26	26	26	26	26	26	26	26
25	26	26	26	26	26	26	26	26	26	26
26	26	26	26	26	26	26	26	26	26	26
27	26	26	26	26	26	26	26	26	26	26
28	26	26	26	26	26	26	NA	26	26	26
29	26	26	26	26	26	26	NA	26	26	26
30	26	26	26	26	26	26	NA	26	26	26
31	NA	NA	NA	26	26	26	NA	NA	NA	NA
32	NA	NA	NA	26	26	NA	NA	NA	NA	NA
33	NA	NA	NA	26	26	NA	NA	NA	NA	NA
34	NA	NA	NA	26	26	NA	NA	NA	NA	NA
35	NA	NA	NA	26	26	NA	NA	NA	NA	NA
36	NA	NA	NA	26	26	NA	NA	NA	NA	NA
37	NA	NA	NA	26	NA	NA	NA	NA	NA	NA
38	NA	NA	NA	26	NA	NA	NA	NA	NA	NA
39	NA	NA	NA	26	NA	NA	NA	NA	NA	NA
40	NA	NA	NA	26	NA	NA	NA	NA	NA	NA
41	NA	NA	NA	25	NA	NA	NA	NA	NA	NA
42	NA	NA	NA	25	NA	NA	NA	NA	NA	NA
43	NA	NA	NA	26	NA	NA	NA	NA	NA	NA
44	NA	NA	NA	26	NA	NA	NA	NA	NA	NA
45	NA	NA	NA	26	NA	NA	NA	NA	NA	NA
46	NA	NA	NA	26	NA	NA	NA	NA	NA	NA

47	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
48	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
49	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
50	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
51	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
52	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
53	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
54	NA	NA	NA	NA	24	NA	NA	NA	NA	NA
55	NA	NA	NA	NA	24	NA	NA	NA	NA	NA
56	NA	NA	NA	NA	24	NA	NA	NA	NA	NA
57	NA	NA	NA	NA	24	NA	NA	NA	NA	NA
58	NA	NA	NA	NA	24	NA	NA	NA	NA	NA
59	NA	NA	NA	NA	24	NA	NA	NA	NA	NA
60	NA	NA	NA	NA	25	NA	NA	NA	NA	NA
61	NA	NA	NA	NA	25	NA	NA	NA	NA	NA
62	NA	NA	NA	NA	25	NA	NA	NA	NA	NA
63	NA	NA	NA	NA	25	24	NA	NA	NA	NA
64	23	NA	NA	NA	25	24	NA	26	NA	NA
65	23	NA	NA	NA	25	24	NA	26	NA	NA
66	23	24	NA	26	25	24	NA	26	24	NA
67	23	24	NA	26	25	25	NA	26	24	NA
68	23	24	NA	24	25	25	NA	26	24	NA
69	23	23	NA	24	25	25	NA	26	24	NA
70	23	23	NA	24	25	25	NA	26	25	NA
71	25	23	NA	24	25	25	NA	26	25	NA
72	25	23	22	24	25	25	NA	25	25	NA
73	25	23	22	24	25	25	NA	25	25	NA
74	25	24	22	25	25	25	NA	25	25	22
75	25	24	24	25	25	25	NA	25	25	22
76	25	24	24	25	25	25	24	25	25	22
77	25	24	24	25	25	25	24	25	25	24
78	24	24	24	25	24	25	24	24	25	24
79	24	24	24	25	24	24	24	24	25	24
80	24	24	25	24	24	24	24	24	25	25
81	25	24	25	24	25	24	24	25	25	25
82	25	25	25	24	25	NA	25	25	25	25
83	25	25	25	24	25	NA	25	25	25	25
84	25	25	25	25	25	NA	25	25	25	25
85	25	25	25	25	25	NA	25	25	25	25
86	25	25	25	25	25	25	25	25	25	25
87	25	25	25	25	25	25	25	25	25	25
88	25	25	25	25	25	25	25	25	25	25
89	25	25	25	25	25	25	25	25	25	25
90	25	25	25	25	25	25	25	25	25	25
91	25	25	25	25	26	25	25	25	25	25
92	25	25	25	25	26	25	25	25	25	25
93	25	25	25	25	26	25	25	25	25	26
94	26	26	25	25	26	26	25	25	26	26
95	26	26	25	25	26	26	25	25	26	26
96	26	26	25	25	26	26	25	25	26	26

Time block	Day 21	Day 22	Day 23	Day 24	Day 25	Day 26	Day 27	Day 28	Day 29	Day 30
1	26	27	27	27	27	27	26	26	26	26
2	26	27	27	27	27	27	26	26	26	26
3	26	27	27	27	27	27	26	26	26	26
4	26	27	27	27	27	27	26	26	26	26
5	26	27	27	27	27	27	26	26	26	26
6	26	27	27	27	27	27	26	26	26	26
7	26	27	27	27	27	27	26	26	26	26
8	26	27	27	27	27	27	26	26	26	26
9	26	27	27	27	27	27	26	26	26	26
10	26	27	27	27	27	27	26	26	26	26
11	26	27	27	27	27	27	26	26	26	26
12	26	27	27	27	27	27	26	26	26	26
13	26	27	27	27	27	27	26	26	26	26
14	26	27	27	27	27	27	26	26	26	26
15	26	27	27	27	27	27	26	26	26	26
16	26	27	27	27	27	27	26	26	26	26
17	26	27	27	27	27	27	26	26	26	26
18	26	27	27	27	27	27	26	26	26	26
19	26	27	27	27	27	27	26	26	26	26
20	26	27	27	27	27	27	26	26	26	26
21	26	27	27	27	27	27	26	26	26	26
22	26	27	27	27	27	27	26	26	26	26
23	26	27	27	27	27	27	26	26	26	26
24	26	27	27	27	27	27	26	26	26	26
25	26	27	27	27	27	27	26	26	26	26
26	26	27	27	27	27	27	26	26	26	26
27	26	27	27	27	27	27	26	26	26	26
28	26	27	27	27	27	27	26	26	26	26
29	25	27	27	27	27	27	26	26	26	26
30	25	27	27	27	27	27	26	26	26	26
31	25	26	26	NA	NA	NA	NA	26	26	NA
32	NA	26	26	NA	NA	NA	NA	26	26	NA
33	NA	26	26	NA	NA	NA	NA	26	26	NA
34	NA	26	26	NA	NA	NA	NA	26	26	NA
35	NA	26	26	NA	NA	NA	NA	26	26	NA
36	NA	26	26	NA	NA	NA	NA	26	26	NA
37	NA	26	26	NA	NA	NA	NA	26	26	NA
38	NA	26	26	NA	NA	NA	NA	26	26	NA
39	NA	26	26	NA	NA	NA	NA	26	26	NA
40	23	26	26	NA	NA	NA	NA	26	26	NA
41	23	26	26	NA	NA	NA	NA	26	26	NA
42	23	26	26	NA	NA	NA	NA	24	26	NA
43	25	26	26	NA	NA	NA	NA	24	26	NA
44	25	26	26	NA	NA	NA	NA	24	26	NA
45	25	26	26	NA	NA	NA	NA	25	26	NA
46	25	26	26	NA	NA	NA	NA	25	26	NA

47	25	26	26	NA	NA	NA	NA	25	26	NA
48	25	26	26	NA	NA	NA	NA	25	26	NA
49	25	26	25	NA	NA	NA	NA	24	25	NA
50	24	26	25	NA	NA	NA	NA	24	25	NA
51	24	26	25	NA	NA	NA	NA	24	25	NA
52	24	26	25	NA	NA	NA	NA	24	25	NA
53	25	26	25	NA	NA	NA	NA	25	25	NA
54	25	26	25	NA	NA	NA	NA	25	25	NA
55	25	26	25	NA	NA	NA	NA	25	24	NA
56	25	26	26	NA	NA	NA	NA	25	24	NA
57	25	26	26	NA	NA	NA	NA	25	24	NA
58	25	26	26	NA	NA	NA	NA	25	24	NA
59	25	26	26	NA	NA	NA	NA	25	24	NA
60	25	26	26	NA	NA	NA	NA	25	24	NA
61	26	26	26	NA	NA	NA	NA	26	24	NA
62	26	26	26	NA	NA	NA	NA	26	25	NA
63	26	26	26	NA	NA	NA	NA	26	25	NA
64	26	26	26	NA	NA	NA	NA	26	25	NA
65	26	26	26	25	NA	NA	NA	26	25	NA
66	26	26	26	25	NA	NA	NA	26	25	24
67	26	26	26	25	NA	NA	NA	26	25	24
68	24	26	26	25	26	NA	NA	24	25	24
69	24	26	26	25	26	NA	NA	24	25	25
70	24	26	26	25	26	25	NA	24	25	25
71	24	26	26	25	26	25	NA	24	25	25
72	25	26	26	25	25	25	NA	25	25	25
73	25	26	26	25	25	25	NA	NA	25	25
74	25	26	26	25	25	25	22	NA	25	25
75	25	26	26	25	25	25	22	NA	25	25
76	25	26	26	25	25	25	22	NA	25	25
77	25	26	26	25	25	25	23	NA	25	25
78	25	26	26	25	25	25	23	NA	25	24
79	25	26	26	25	25	NA	23	NA	25	24
80	24	26	26	25	25	NA	25	NA	25	24
81	24	26	26	25	25	NA	25	NA	25	24
82	24	26	26	25	25	24	25	NA	25	26
83	25	26	26	25	25	24	25	NA	25	26
84	25	26	26	25	25	24	25	NA	25	26
85	25	26	26	25	25	25	25	NA	25	26
86	25	26	25	25	25	25	25	NA	25	25
87	25	26	25	25	25	25	25	NA	25	25
88	25	26	25	25	25	25	25	23	25	25
89	25	26	25	25	25	25	25	23	25	25
90	25	26	26	25	25	25	25	23	25	26
91	25	26	26	25	25	25	25	25	25	26
92	25	26	26	25	25	25	25	25	25	26
93	25	27	26	25	27	25	25	25	25	26
94	25	27	26	25	27	26	25	25	26	26
95	25	27	26	25	27	26	25	25	26	26
96	25	27	26	25	27	26	25	25	26	26

Time block	Day 31	Day 32	Day 33	Day 34	Day 35	Day 36	Day 37	Day 38	Day 39	Day 40
1	26	26	26	26	26	25	25	26	26	26
2	26	26	26	26	26	25	25	26	26	26
3	26	26	26	26	26	26	25	26	26	26
4	26	26	26	26	26	26	25	26	26	26
5	26	26	26	26	26	26	25	26	26	26
6	26	26	26	26	26	26	25	26	26	26
7	26	26	26	26	26	26	26	26	26	26
8	26	26	26	26	26	26	26	26	26	26
9	26	26	26	26	26	26	26	26	26	26
10	26	26	26	26	26	26	26	26	26	26
11	26	26	26	26	26	26	26	26	26	26
12	26	26	26	26	26	26	26	26	26	26
13	26	26	26	26	26	26	26	26	26	26
14	26	26	26	26	26	26	26	26	26	26
15	26	26	26	26	26	26	26	26	26	26
16	26	26	26	26	26	26	26	26	27	26
17	26	26	26	26	26	26	26	26	27	26
18	26	26	26	26	26	26	26	26	27	26
19	26	26	26	26	26	26	26	26	27	26
20	26	26	26	26	26	26	26	26	27	26
21	26	26	26	26	26	26	26	26	27	26
22	26	26	26	26	26	26	26	26	27	26
23	26	26	26	26	26	26	26	26	27	26
24	26	26	26	26	26	26	26	26	27	26
25	26	26	26	26	26	26	26	26	27	26
26	26	26	26	26	26	26	26	26	NA	26
27	26	NA	26	26	26	26	26	26	NA	26
28	26	NA	26	26	26	26	26	26	NA	26
29	26	NA	NA	26	26	26	NA	26	NA	NA
30	26	NA	NA	26	26	26	NA	NA	NA	NA
31	NA	NA	NA	26	26	26	NA	NA	NA	NA
32	NA	NA	NA	26	26	26	NA	NA	NA	NA
33	NA	NA	NA	NA	26	26	NA	NA	NA	NA
34	NA	NA	NA	NA	26	26	NA	NA	NA	NA
35	NA	NA	NA	NA	26	26	NA	NA	NA	NA
36	NA	NA	NA	NA	26	26	NA	NA	NA	NA
37	NA	NA	NA	NA	26	26	NA	NA	NA	NA
38	NA	NA	NA	NA	26	26	NA	NA	NA	NA
39	NA	NA	NA	NA	26	26	NA	NA	NA	NA
40	NA	NA	NA	NA	26	25	NA	NA	NA	NA
41	NA	NA	NA	NA	25	25	NA	NA	NA	NA
42	NA	NA	NA	NA	25	24	NA	NA	NA	NA
43	NA	NA	NA	NA	NA	24	NA	NA	NA	NA
44	NA	NA	NA	NA	NA	24	NA	NA	NA	NA
45	NA	NA	NA	NA	NA	24	NA	NA	NA	NA
46	NA	NA	NA	NA	NA	24	NA	NA	NA	NA

47	NA	NA	NA	NA	NA	24	NA	NA	NA	NA
48	NA	NA	NA	NA	NA	24	NA	NA	NA	NA
49	NA	NA	NA	NA	NA	24	NA	NA	NA	NA
50	NA	NA	NA	NA	NA	24	NA	NA	NA	NA
51	NA	NA	NA	NA	NA	24	NA	NA	NA	NA
52	NA	NA	NA	NA	NA	24	NA	NA	NA	NA
53	NA	NA	NA	NA	NA	24	NA	NA	NA	NA
54	NA	NA	NA	NA	NA	24	NA	NA	NA	NA
55	NA	NA	NA	NA	NA	24	NA	NA	NA	NA
56	NA	NA	NA	NA	NA	24	NA	NA	NA	NA
57	NA	NA	NA	NA	NA	24	NA	NA	NA	NA
58	NA	NA	NA	NA	NA	24	NA	NA	NA	NA
59	NA	NA	NA	NA	NA	24	NA	NA	NA	NA
60	NA	NA	NA	NA	NA	25	NA	NA	NA	NA
61	NA	NA	NA	NA	NA	25	NA	NA	NA	24
62	NA	NA	NA	NA	NA	25	NA	NA	NA	24
63	NA	NA	NA	NA	NA	25	NA	24	NA	24
64	NA	NA	NA	NA	NA	25	NA	24	NA	24
65	NA	NA	NA	NA	NA	25	NA	24	NA	24
66	NA	NA	24	NA	NA	25	NA	24	23	24
67	NA	NA	24	NA	NA	25	NA	24	23	24
68	NA	NA	24	NA	NA	25	NA	24	23	24
69	NA	NA	24	NA	23	25	25	24	23	24
70	NA	NA	24	NA	23	NA	25	24	23	24
71	NA	NA	24	NA	23	NA	25	24	23	24
72	25	NA	24	NA	23	NA	25	24	23	25
73	25	NA	25	NA	23	NA	25	24	NA	25
74	25	NA	25	NA	24	NA	25	24	NA	25
75	25	24	25	22	24	NA	25	24	NA	25
76	25	24	25	22	24	NA	25	24	NA	25
77	25	24	25	22	24	NA	25	24	NA	25
78	25	24	25	22	24	NA	25	24	NA	25
79	25	24	25	22	24	NA	NA	24	NA	25
80	25	24	25	22	24	NA	NA	24	23	25
81	25	25	25	24	24	NA	NA	24	23	25
82	25	25	25	24	25	NA	NA	24	23	25
83	25	25	25	24	25	NA	NA	24	23	25
84	25	25	25	24	25	23	24	24	23	25
85	25	25	25	24	25	23	24	25	23	25
86	25	25	25	24	25	23	24	25	23	25
87	25	25	25	24	25	23	24	25	23	25
88	25	25	25	24	25	23	24	25	24	25
89	25	25	25	24	25	23	25	25	24	25
90	25	25	25	24	25	24	25	25	24	25
91	25	25	25	24	25	24	25	25	24	25
92	25	25	25	25	25	24	25	25	24	25
93	25	25	25	25	25	24	25	25	24	25
94	25	25	25	25	25	24	25	25	26	25
95	26	25	26	25	25	24	26	25	26	25
96	26	25	26	26	25	24	26	25	26	25

Time block	Day 41	Day 42	Day 43	Day 44	Day 45	Day 46	Day 47	Day 48	Day 49	Day 50
1	25	25	25	24	26	26	26	25	25	25
2	25	25	25	24	26	26	26	25	25	26
3	25	25	25	25	26	26	26	26	25	26
4	26	25	26	25	26	26	26	26	25	26
5	26	25	26	25	26	26	26	26	25	26
6	26	25	26	25	26	26	26	26	25	26
7	26	26	26	25	26	26	26	26	26	26
8	26	26	26	25	26	26	26	26	26	26
9	26	26	26	25	26	26	26	26	26	26
10	26	26	26	25	26	26	26	26	26	26
11	26	26	26	25	26	26	26	26	26	26
12	26	26	26	25	26	26	26	26	26	26
13	26	26	26	25	26	26	26	26	26	26
14	26	26	26	25	26	26	26	26	26	26
15	26	26	26	25	26	26	26	26	26	26
16	26	26	26	25	26	26	26	26	26	26
17	26	26	26	25	26	26	26	26	26	26
18	26	26	26	25	26	26	26	26	26	26
19	26	26	26	25	26	26	26	26	26	26
20	26	26	26	25	26	26	26	26	26	26
21	26	26	26	25	26	26	26	26	26	26
22	26	26	26	25	26	26	26	26	26	26
23	26	26	26	25	26	26	26	26	26	26
24	26	26	26	25	26	26	26	26	26	26
25	26	26	26	25	26	26	NA	26	26	26
26	NA	26	26	25	26	26	NA	26	26	26
27	NA	26	26	25	26	26	NA	26	26	26
28	NA	26	26	25	26	26	NA	26	26	26
29	NA	26	26	25	26	26	NA	26	26	26
30	NA	26	26	25	26	NA	NA	NA	26	26
31	NA	26	26	NA	26	NA	NA	NA	26	26
32	NA	26	26	NA	26	NA	NA	NA	26	26
33	NA	26	26	NA	26	NA	NA	NA	26	26
34	NA	26	26	NA	26	NA	NA	NA	26	26
35	NA	26	26	NA	26	NA	NA	NA	26	26
36	NA	26	26	NA	NA	NA	NA	NA	26	26
37	NA	26	26	NA	NA	NA	NA	NA	26	26
38	NA	26	26	NA	NA	NA	NA	NA	26	26
39	NA	26	26	NA	NA	NA	NA	NA	26	26
40	NA	26	26	NA	NA	NA	NA	NA	26	26
41	NA	25	26	NA	NA	NA	NA	NA	25	26
42	NA	25	26	NA	NA	NA	NA	NA	25	26
43	NA	26	26	NA	NA	NA	NA	NA	26	26
44	NA	26	26	NA	NA	NA	NA	NA	26	26
45	NA	26	26	NA	NA	NA	NA	NA	26	26
46	NA	26	26	NA	NA	NA	NA	NA	26	26

47	NA	26	26	NA	NA	NA	NA	NA	26	26
48	NA	NA	26	NA	NA	NA	NA	NA	26	26
49	NA	NA	26	NA	NA	NA	NA	NA	26	26
50	NA	NA	26	NA	NA	NA	NA	NA	NA	26
51	NA	NA	26	NA	NA	NA	NA	NA	NA	26
52	NA	NA	26	NA	NA	NA	NA	NA	NA	26
53	NA	NA	26	NA	NA	NA	NA	NA	NA	26
54	NA	NA	26	NA	NA	NA	NA	NA	NA	26
55	NA	NA	24	NA	NA	NA	NA	NA	NA	26
56	NA	NA	24	NA	NA	NA	NA	NA	NA	26
57	NA	NA	24	NA	NA	NA	NA	NA	26	26
58	NA	NA	24	NA	NA	NA	NA	NA	26	26
59	NA	NA	24	NA	NA	NA	NA	NA	26	26
60	NA	NA	24	NA	NA	NA	NA	NA	26	26
61	NA	NA	24	NA	NA	NA	NA	NA	26	26
62	NA	NA	24	NA	NA	NA	NA	NA	26	26
63	NA	NA	24	25	NA	NA	NA	NA	26	26
64	NA	NA	24	25	NA	NA	NA	NA	26	26
65	NA	23	24	25	NA	NA	NA	NA	26	26
66	NA	23	24	25	NA	25	24	NA	23	26
67	NA	23	24	25	NA	25	24	NA	23	26
68	NA	23	24	25	NA	25	24	NA	23	26
69	NA	23	24	25	NA	25	24	NA	23	26
70	NA	23	24	25	NA	NA	24	NA	23	26
71	NA	23	24	25	NA	NA	24	NA	23	26
72	NA	23	24	25	NA	NA	24	23	23	26
73	NA	23	24	25	NA	NA	24	23	NA	26
74	NA	25	24	25	24	NA	24	23	NA	26
75	NA	25	24	25	24	25	24	23	NA	26
76	24	25	24	25	24	25	24	23	NA	26
77	24	25	NA	25	24	25	25	23	NA	26
78	24	25	NA	25	24	25	25	23	NA	26
79	24	25	NA	25	24	25	25	23	NA	26
80	24	25	NA	25	24	25	25	23	NA	26
81	24	24	NA	25	25	25	25	23	NA	26
82	24	24	NA	25	25	25	25	23	NA	26
83	24	24	NA	25	25	25	25	25	NA	26
84	24	24	NA	25	25	25	25	25	NA	26
85	24	24	NA	25	25	25	25	25	24	26
86	25	24	23	25	25	25	25	25	24	26
87	25	24	23	25	25	25	25	25	24	26
88	25	24	23	25	25	25	25	25	24	26
89	25	24	23	25	25	25	25	25	24	26
90	25	24	23	25	25	25	25	25	24	26
91	25	25	23	25	25	25	25	25	25	26
92	25	25	23	25	25	25	25	25	25	26
93	25	25	24	25	25	25	25	25	25	26
94	25	25	24	25	25	26	25	25	25	26
95	25	25	24	26	25	26	25	25	25	26
96	25	25	24	26	25	26	25	25	25	26

Time block	Day 51	Day 52	Day 53	Day 54	Day 55	Day 56	Day 57	Day 58	Day 59	Day 60
1	26	26	26	26	25	25	26	26	26	26
2	26	26	26	26	25	25	26	26	26	26
3	26	26	26	26	26	26	26	26	26	26
4	26	26	26	26	26	26	26	26	26	26
5	26	26	26	26	26	26	26	26	26	26
6	26	26	26	26	26	26	26	26	26	26
7	26	26	26	26	26	26	26	26	26	26
8	26	26	26	26	26	26	26	26	26	26
9	26	26	26	26	26	26	26	26	26	26
10	26	26	26	26	26	26	26	26	26	26
11	26	26	26	26	26	26	26	26	26	26
12	26	26	26	26	26	26	26	26	26	26
13	26	26	26	26	26	26	26	26	26	26
14	26	26	26	26	26	26	26	26	26	26
15	26	26	26	26	26	26	26	26	26	26
16	26	26	26	26	26	26	26	26	26	26
17	26	26	26	26	26	26	26	26	26	26
18	26	26	26	26	26	26	26	26	26	26
19	26	26	26	26	26	26	26	26	26	26
20	26	26	26	26	26	26	26	26	26	26
21	26	26	26	26	26	26	26	26	26	26
22	26	26	26	26	26	26	26	26	26	26
23	26	26	26	26	26	26	26	26	26	26
24	26	26	26	26	26	26	26	26	26	26
25	26	26	26	26	26	26	26	26	26	26
26	26	26	26	26	26	26	26	26	26	26
27	26	26	26	26	26	26	26	26	26	26
28	26	26	26	26	26	26	26	26	26	26
29	26	26	26	26	26	26	26	26	26	26
30	26	26	26	26	26	26	26	26	26	26
31	26	26	NA	26	NA	26	26	NA	26	26
32	26	26	NA	NA	NA	26	26	NA	26	26
33	NA	26	NA	NA	NA	26	26	NA	26	NA
34	NA	26	NA	NA	NA	26	26	NA	26	NA
35	NA	NA	NA	NA	NA	26	26	NA	26	NA
36	NA	NA	NA	NA	NA	26	26	NA	26	NA
37	NA	NA	NA	NA	NA	26	26	NA	26	NA
38	NA	NA	NA	NA	NA	26	26	NA	26	NA
39	NA	NA	NA	NA	NA	26	26	NA	26	NA
40	NA	NA	NA	NA	NA	26	26	NA	26	NA
41	NA	NA	NA	NA	NA	25	26	NA	26	NA
42	NA	NA	NA	NA	NA	25	26	NA	26	NA
43	NA	NA	NA	NA	NA	26	26	NA	25	NA
44	NA	NA	NA	NA	NA	26	26	NA	25	NA
45	NA	NA	NA	NA	NA	26	26	NA	NA	NA
46	NA	NA	NA	NA	NA	26	26	NA	NA	NA

47	NA	NA	NA	NA	NA	26	26	NA	NA	NA
48	NA	NA	NA	NA	NA	26	26	NA	NA	NA
49	NA	NA	NA	NA	NA	26	26	NA	NA	NA
50	NA	NA	NA	NA	NA	NA	26	NA	NA	NA
51	NA	NA	NA	NA	NA	NA	26	NA	NA	NA
52	NA	NA	NA	NA	NA	NA	26	NA	NA	NA
53	NA	NA	NA	NA	NA	NA	26	NA	NA	NA
54	NA	NA	NA	NA	NA	NA	26	NA	NA	NA
55	NA	NA	NA	NA	NA	NA	26	NA	NA	NA
56	NA	NA	NA	NA	NA	NA	26	NA	NA	NA
57	NA	NA	NA	NA	NA	NA	26	NA	24	NA
58	NA	NA	NA	NA	NA	NA	26	NA	24	NA
59	NA	NA	NA	NA	NA	NA	26	NA	24	NA
60	25	NA	NA	NA	NA	NA	26	NA	24	NA
61	25	NA	NA	NA	NA	NA	26	NA	24	NA
62	25	NA	NA	NA	NA	NA	26	24	24	NA
63	25	NA	NA	NA	NA	NA	26	24	24	NA
64	25	NA	NA	24	NA	NA	26	24	24	NA
65	25	NA	NA	24	23	24	26	24	24	24
66	25	NA	NA	24	23	24	26	24	24	24
67	25	NA	NA	24	23	24	26	24	24	24
68	25	NA	NA	25	23	24	NA	24	24	24
69	NA	24	25	25	23	24	NA	24	24	24
70	NA	24	25	25	23	24	NA	24	24	24
71	NA	24	25	25	23	24	NA	NA	24	24
72	NA	24	25	25	23	24	NA	NA	25	25
73	NA	24	25	25	23	24	NA	NA	25	25
74	NA	24	25	25	24	24	NA	NA	25	25
75	24	24	25	25	24	25	NA	NA	25	25
76	24	24	25	25	24	25	NA	NA	25	25
77	24	24	25	25	24	25	NA	NA	25	25
78	24	24	NA	25	24	25	NA	NA	25	25
79	24	24	NA	25	24	25	24	24	25	25
80	24	24	NA	25	24	25	24	24	25	25
81	24	25	NA	25	24	25	24	24	25	25
82	24	25	24	25	24	25	24	24	25	25
83	24	25	24	25	24	25	24	24	25	25
84	25	25	24	25	24	25	24	24	25	25
85	25	25	24	25	25	25	24	24	25	25
86	25	25	24	25	25	25	24	24	25	25
87	25	25	24	25	25	25	24	24	25	25
88	25	25	24	25	25	25	24	25	25	25
89	25	25	25	25	25	25	24	25	25	25
90	25	25	25	25	25	25	24	25	25	25
91	25	25	25	25	25	25	24	25	25	25
92	25	25	25	25	25	25	26	25	25	25
93	25	25	25	25	25	25	26	25	25	25
94	25	25	25	25	25	25	26	25	25	25
95	26	25	25	25	25	25	26	25	25	25
96	26	25	26	25	25	25	26	25	25	26

A retired man's schedule

Time block	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
1	28	28	28	28	28	28	28	28	28	NA
2	28	28	28	28	28	28	28	28	28	NA
3	28	28	28	28	28	28	28	28	28	NA
4	28	28	28	28	28	28	28	28	28	NA
5	28	28	28	28	28	28	28	28	28	NA
6	28	28	28	28	28	28	28	28	28	NA
7	28	28	28	28	28	28	28	28	28	NA
8	28	28	28	28	28	28	28	28	28	NA
9	28	28	28	28	28	28	28	28	28	NA
10	28	28	28	28	28	28	28	28	28	NA
11	28	28	28	28	28	28	28	28	28	NA
12	28	28	28	28	28	28	28	28	28	NA
13	28	28	28	28	28	28	28	28	28	NA
14	28	28	28	28	28	28	28	28	28	NA
15	28	28	28	28	28	28	28	28	28	NA
16	28	28	28	28	28	28	28	28	28	NA
17	28	28	28	28	28	28	28	28	28	NA
18	28	28	28	28	28	28	28	28	28	NA
19	28	28	28	28	28	28	28	28	28	NA
20	28	28	28	28	28	28	28	28	28	NA
21	28	28	28	28	28	28	28	28	28	NA
22	28	28	28	28	28	28	28	28	28	NA
23	28	28	28	28	28	28	28	28	28	NA
24	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
25	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
26	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
27	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
28	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
29	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
30	28	NA	NA	NA	28	NA	NA	28	NA	NA
31	28	NA	28	NA	28	NA	28	28	NA	NA
32	28	28	28	NA	28	28	28	28	28	NA
33	28	28	28	NA	28	28	28	28	28	NA
34	28	28	28	28	28	28	28	28	28	NA
35	28	28	28	28	28	28	28	28	28	NA
36	28	28	28	28	28	28	28	28	28	NA
37	28	28	28	28	28	28	28	28	28	NA
38	28	28	28	28	28	28	28	28	28	NA
39	28	28	28	28	28	28	28	28	28	NA
40	28	28	28	28	28	28	28	28	28	NA
41	28	28	28	28	28	28	28	28	28	NA
42	28	28	28	28	28	28	28	28	28	NA
43	28	28	28	28	28	28	28	28	28	NA
44	28	28	28	28	28	28	28	28	28	NA
45	28	28	28	28	28	28	28	28	28	NA
46	27	27	27	27	27	27	27	27	27	NA

47	27	27	27	27	27	27	27	27	27	NA
48	27	27	27	27	27	27	27	27	27	NA
49	27	27	27	27	27	27	27	27	27	NA
50	27	27	27	27	27	27	27	27	27	NA
51	28	28	28	28	28	28	28	28	28	NA
52	28	28	28	28	28	28	28	28	28	NA
53	28	28	28	28	28	28	28	28	28	NA
54	28	28	28	28	28	28	28	28	28	NA
55	28	28	28	28	28	28	28	28	28	NA
56	28	28	28	28	28	28	28	28	28	NA
57	28	28	28	28	28	28	28	28	28	NA
58	28	28	28	28	28	28	28	28	28	NA
59	28	28	28	28	28	28	28	28	28	NA
60	27	27	27	27	27	27	27	27	27	NA
61	27	27	27	27	27	27	NA	27	27	NA
62	27	27	27	27	27	27	NA	27	27	NA
63	27	27	27	27	27	27	NA	27	27	NA
64	27	27	27	27	27	27	NA	27	27	NA
65	27	27	27	27	27	27	NA	27	27	NA
66	27	27	27	27	27	27	NA	27	27	NA
67	27	27	27	27	27	27	NA	27	27	NA
68	27	27	27	27	27	27	NA	27	27	NA
69	27	27	27	27	27	27	NA	27	27	NA
70	27	27	27	27	27	27	27	27	27	NA
71	27	27	27	27	27	27	27	27	27	NA
72	27	27	27	27	27	27	27	27	27	NA
73	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
74	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
75	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
76	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
77	28	28	28	28	28	28	28	28	28	NA
78	28	28	28	28	28	28	28	28	28	NA
79	28	28	28	28	28	28	28	28	28	NA
80	28	28	28	28	28	28	28	28	28	NA
81	28	28	28	28	28	28	28	28	28	NA
82	28	28	28	28	28	28	28	28	28	NA
83	28	28	28	28	28	28	28	28	28	NA
84	28	28	28	28	28	28	28	28	28	NA
85	28	28	28	28	28	28	28	28	28	NA
86	28	28	28	28	28	28	28	28	28	NA
87	28	28	28	28	28	28	28	28	28	NA
88	28	28	28	28	28	28	28	28	28	NA
89	28	28	28	28	28	28	28	28	28	NA
90	28	28	28	28	28	28	28	28	28	NA
91	28	28	28	28	28	28	28	28	28	NA
92	28	28	28	28	28	28	28	28	28	NA
93	28	28	28	28	28	28	28	28	28	NA
94	28	28	28	28	28	28	28	28	28	NA
95	28	28	28	28	28	28	28	28	28	NA
96	28	28	28	28	28	28	28	28	28	NA

Time block	Day 11	Day 12	Day 13	Day 14	Day 15	Day 16	Day 17	Day 18	Day 19	Day 20
1	NA	NA	NA	NA	25	25	26	27	27	27
2	NA	NA	NA	NA	25	25	26	27	27	27
3	NA	NA	NA	NA	25	25	26	27	27	27
4	NA	NA	NA	NA	25	25	26	27	27	27
5	NA	NA	NA	NA	25	25	26	27	27	27
6	NA	NA	NA	NA	25	25	26	27	27	27
7	NA	NA	NA	NA	25	25	26	27	27	27
8	NA	NA	NA	NA	25	25	26	27	27	27
9	NA	NA	NA	NA	25	25	26	27	27	27
10	NA	NA	NA	NA	25	25	26	27	27	27
11	NA	NA	NA	NA	25	25	26	27	27	27
12	NA	NA	NA	NA	25	25	26	27	27	27
13	NA	NA	NA	NA	25	25	26	27	27	27
14	NA	NA	NA	NA	25	25	26	27	27	27
15	NA	NA	NA	NA	25	25	26	27	27	27
16	NA	NA	NA	NA	25	25	26	27	27	27
17	NA	NA	NA	NA	25	25	26	27	27	27
18	NA	NA	NA	NA	25	25	26	27	27	27
19	NA	NA	NA	NA	25	25	26	27	27	27
20	NA	NA	NA	NA	25	25	26	27	27	27
21	NA	NA	NA	NA	25	25	26	27	27	27
22	NA	NA	NA	NA	25	25	26	27	27	27
23	NA	NA	NA	NA	25	25	26	27	27	27
24	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
25	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
26	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
27	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
28	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
29	NA	NA	NA	NA	NA	NA	26	NA	NA	NA
30	NA	NA	NA	NA	NA	NA	26	NA	NA	NA
31	NA	NA	NA	NA	25	NA	26	NA	NA	NA
32	NA	NA	NA	NA	25	25	26	NA	NA	NA
33	NA	NA	NA	NA	25	25	26	27	NA	27
34	NA	NA	NA	NA	25	25	26	27	27	27
35	NA	NA	NA	NA	25	25	26	27	27	27
36	NA	NA	NA	NA	25	25	26	27	27	27
37	NA	NA	NA	NA	25	25	26	27	27	27
38	NA	NA	NA	NA	25	25	26	27	27	27
39	NA	NA	NA	NA	25	25	26	27	27	27
40	NA	NA	NA	NA	25	25	26	27	27	27
41	NA	NA	NA	NA	25	25	26	27	27	27
42	NA	NA	NA	NA	25	25	26	27	27	27
43	NA	NA	NA	NA	25	25	26	27	27	27
44	NA	NA	NA	NA	25	25	26	27	27	27
45	NA	NA	NA	NA	25	25	26	27	27	27
46	NA	NA	NA	NA	25	25	26	27	27	27

47	NA	NA	NA	NA	25	25	26	27	27	27
48	NA	NA	NA	NA	25	25	26	27	27	27
49	NA	NA	NA	NA	25	25	26	27	27	27
50	NA	NA	NA	NA	25	25	26	27	27	27
51	NA	NA	NA	NA	25	25	26	27	27	27
52	NA	NA	NA	NA	25	25	26	27	27	27
53	NA	NA	NA	NA	25	25	26	27	27	27
54	NA	NA	NA	NA	25	25	26	27	27	27
55	NA	NA	NA	NA	25	25	26	27	27	27
56	NA	NA	NA	NA	25	25	26	27	27	27
57	NA	NA	NA	NA	25	25	26	27	27	27
58	NA	NA	NA	NA	25	25	26	27	27	27
59	NA	NA	NA	NA	25	25	26	27	27	27
60	NA	NA	NA	NA	25	25	26	27	27	27
61	NA	NA	NA	NA	25	25	26	27	27	27
62	NA	NA	NA	NA	25	25	26	27	27	27
63	NA	NA	NA	NA	25	25	26	27	27	27
64	NA	NA	NA	NA	25	25	26	27	27	27
65	NA	NA	NA	NA	25	25	NA	27	27	27
66	NA	NA	NA	NA	25	25	NA	27	27	27
67	NA	NA	NA	NA	25	25	NA	27	27	27
68	NA	NA	NA	NA	25	25	NA	27	27	27
69	NA	NA	NA	NA	25	25	NA	27	27	27
70	NA	NA	NA	NA	25	25	NA	27	27	27
71	NA	NA	NA	NA	25	25	NA	27	27	27
72	NA	NA	NA	NA	25	25	26	27	27	27
73	NA	NA	NA	NA	NA	NA	26	NA	NA	NA
74	NA	NA	NA	NA	NA	NA	26	NA	NA	NA
75	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
76	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
77	NA	NA	NA	NA	25	25	26	27	27	27
78	NA	NA	NA	NA	25	25	26	27	27	27
79	NA	NA	NA	NA	25	25	26	27	27	27
80	NA	NA	NA	NA	25	25	26	27	27	27
81	NA	NA	NA	NA	25	25	26	27	27	27
82	NA	NA	NA	NA	25	25	26	27	27	27
83	NA	NA	NA	NA	25	25	26	27	27	27
84	NA	NA	NA	NA	25	25	26	27	27	27
85	NA	NA	NA	NA	25	25	26	27	27	27
86	NA	NA	NA	NA	25	26	26	27	27	27
87	NA	NA	NA	NA	25	26	26	27	27	27
88	NA	NA	NA	NA	25	26	26	27	27	27
89	NA	NA	NA	NA	25	26	26	27	27	27
90	NA	NA	NA	NA	25	26	26	27	27	27
91	NA	NA	NA	NA	25	26	26	27	27	27
92	NA	NA	NA	NA	25	26	26	27	27	27
93	NA	NA	NA	NA	25	26	26	27	27	27
94	NA	NA	NA	NA	25	26	26	27	27	27
95	NA	NA	NA	NA	25	26	26	27	27	27
96	NA	NA	NA	NA	25	26	26	27	27	27

Time block	Day 21	Day 22	Day 23	Day 24	Day 25	Day 26	Day 27	Day 28	Day 29	Day 30
1	25	23	23	23	23	22	22	22	22	21
2	25	23	23	23	23	22	22	22	22	21
3	25	23	23	23	23	22	22	22	22	21
4	25	23	23	23	23	22	22	22	22	21
5	25	23	23	23	23	22	22	22	22	21
6	25	23	23	23	23	22	22	22	22	21
7	25	23	23	23	23	22	22	22	22	21
8	25	23	23	23	23	22	22	22	22	21
9	25	23	23	23	23	22	22	22	22	21
10	25	23	23	23	23	22	22	22	22	21
11	25	23	23	23	23	22	22	22	22	21
12	25	23	23	23	23	22	22	22	22	21
13	25	23	23	23	23	22	22	22	22	21
14	25	23	23	23	23	22	22	22	22	21
15	25	23	23	23	23	22	22	22	22	21
16	25	23	23	23	23	22	22	22	22	21
17	25	23	23	23	23	22	22	22	22	21
18	25	23	23	23	23	22	22	22	22	21
19	25	23	23	23	23	22	22	22	22	21
20	25	23	23	23	23	22	22	22	22	21
21	25	23	23	23	23	22	22	22	22	21
22	25	23	23	23	23	22	22	22	22	21
23	25	23	23	23	23	22	22	22	22	21
24	NA	NA	NA	NA	NA	NA	NA	NA	NA	21
25	NA	NA	NA	NA	NA	NA	NA	NA	NA	21
26	NA	NA	NA	NA	NA	NA	NA	NA	NA	21
27	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
28	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
29	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
30	NA	NA	NA	NA	NA	22	NA	NA	NA	NA
31	NA	NA	NA	NA	NA	22	NA	NA	NA	NA
32	NA	NA	23	NA	22	22	NA	NA	NA	NA
33	27	23	23	NA	22	22	22	NA	23	NA
34	27	23	23	23	22	22	22	NA	23	NA
35	27	23	23	23	22	22	22	22	23	21
36	27	23	23	23	22	22	22	22	23	21
37	27	23	23	23	22	22	22	22	23	21
38	27	23	23	23	22	22	22	22	23	21
39	27	23	NA	23	22	22	22	22	23	21
40	27	23	NA	23	22	22	22	22	23	21
41	27	23	NA	23	22	22	22	22	23	21
42	27	23	NA	23	22	22	22	22	23	21
43	27	23	NA	23	22	22	22	22	23	21
44	27	23	NA	23	22	22	22	22	23	21
45	27	23	24	23	22	22	22	22	23	23
46	27	23	24	23	22	22	22	22	23	23

47	27	23	24	23	22	22	22	22	23	23
48	27	23	24	23	22	22	22	22	23	23
49	27	23	23	23	22	22	22	22	23	23
50	27	23	23	23	22	22	22	22	23	23
51	27	23	23	23	22	22	22	22	23	23
52	27	23	23	23	22	22	22	22	23	23
53	27	23	23	23	22	22	22	22	23	23
54	27	23	23	23	22	22	22	22	23	23
55	27	23	23	23	22	22	22	22	23	23
56	27	23	23	23	22	22	22	22	23	23
57	27	23	23	23	22	22	22	22	23	23
58	27	23	23	23	22	22	22	22	23	23
59	27	23	23	23	22	22	22	22	23	23
60	27	23	23	23	22	22	22	22	23	23
61	NA	23	23	23	22	22	22	22	23	23
62	NA	23	23	23	22	22	22	NA	23	23
63	NA	23	23	23	22	22	22	NA	23	23
64	NA	23	23	23	22	22	22	NA	23	23
65	NA	23	23	23	22	22	22	NA	23	23
66	NA	23	23	23	22	22	22	NA	23	23
67	27	23	23	23	22	22	22	22	23	23
68	27	23	23	23	22	22	22	22	23	23
69	27	23	23	23	22	22	22	22	23	23
70	27	23	23	23	22	22	22	22	23	23
71	27	23	23	23	22	22	22	22	23	23
72	27	23	23	23	22	22	22	22	23	23
73	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
74	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
75	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
76	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
77	27	24	23	23	22	22	22	22	22	21
78	27	24	23	23	22	22	22	22	22	21
79	27	24	23	23	22	22	22	22	22	21
80	27	24	23	23	22	22	22	22	22	21
81	27	24	23	23	22	22	22	22	22	21
82	27	24	23	23	22	22	22	22	22	21
83	27	24	23	23	22	22	22	22	22	21
84	27	24	23	23	22	22	22	22	22	21
85	27	24	23	23	22	22	22	22	22	21
86	27	23	23	23	22	22	22	22	22	21
87	27	23	23	23	22	22	22	22	23	21
88	27	23	23	23	22	22	22	22	23	21
89	27	23	23	23	22	22	22	22	23	21
90	27	23	23	23	22	22	22	22	23	21
91	27	23	23	23	22	22	22	22	23	21
92	27	23	23	23	22	22	22	22	23	21
93	27	23	23	23	22	22	22	22	23	21
94	27	23	23	23	22	22	22	22	23	21
95	27	23	23	23	22	22	22	22	23	21
96	27	23	23	23	22	22	22	22	23	21

Time block	Day 31	Day 32	Day 33	Day 34	Day 35	Day 36	Day 37	Day 38	Day 39	Day 40
1	21	23	22	23	23	23	22	21	21	20
2	21	23	22	23	23	23	22	21	21	20
3	21	23	22	23	23	23	22	21	21	20
4	21	23	22	23	23	23	22	21	21	20
5	21	23	22	23	23	23	22	21	21	20
6	21	23	22	23	23	23	22	21	21	20
7	21	23	22	23	23	23	22	21	21	20
8	21	23	22	23	23	23	22	21	21	20
9	21	23	22	23	23	23	22	21	21	20
10	21	23	22	23	23	23	22	21	21	20
11	21	23	22	23	23	23	22	21	21	20
12	21	23	22	23	23	23	22	21	21	20
13	21	23	22	23	23	23	22	21	21	20
14	21	23	22	23	23	23	22	21	21	20
15	21	23	22	23	23	23	22	21	21	20
16	21	23	22	23	23	23	22	21	21	20
17	21	23	22	23	23	23	22	21	21	20
18	21	23	22	23	23	23	22	21	21	20
19	21	23	22	23	23	23	22	21	21	20
20	21	23	22	23	23	23	22	21	21	20
21	21	23	22	23	23	23	22	21	21	20
22	21	23	22	23	23	23	22	21	21	20
23	21	23	22	23	23	23	22	21	21	20
24	NA	NA	NA	NA	NA	NA	22	NA	NA	NA
25	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
26	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
27	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
28	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
29	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
30	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
31	NA	NA	22	NA	NA	NA	21	20	NA	20
32	NA	NA	22	NA	NA	NA	21	20	20	20
33	NA	22	22	23	23	21	21	20	20	20
34	21	22	22	23	23	21	21	20	20	20
35	21	22	22	23	23	21	21	20	20	20
36	21	22	22	23	23	21	21	20	20	20
37	21	22	22	23	23	21	21	20	20	20
38	21	22	22	23	23	21	21	20	20	20
39	21	22	22	23	23	21	21	20	20	20
40	21	22	22	23	23	21	21	20	20	20
41	21	22	22	23	23	21	21	20	20	20
42	23	22	22	23	23	21	21	20	20	20
43	23	22	22	23	23	21	21	20	20	20
44	23	22	22	23	23	21	21	20	20	20
45	23	22	22	23	23	21	21	20	20	20
46	23	22	22	23	23	21	21	20	20	20

47	23	22	22	23	23	21	21	20	20	20
48	23	22	22	23	23	21	21	20	20	20
49	23	22	22	23	23	21	21	20	20	20
50	23	22	22	23	23	21	21	20	20	20
51	23	22	22	23	23	21	21	20	20	20
52	23	22	22	23	23	21	21	20	20	20
53	23	22	22	23	23	21	21	20	20	20
54	23	22	22	23	23	21	21	20	20	20
55	23	22	22	23	23	21	21	20	20	20
56	23	22	22	23	23	21	21	20	20	20
57	23	22	22	23	23	21	21	20	20	20
58	23	22	22	23	23	21	21	20	20	20
59	23	22	22	23	23	21	21	20	20	20
60	23	22	22	23	23	21	21	20	20	20
61	23	22	22	23	23	21	21	20	20	NA
62	23	22	22	23	23	21	21	21	20	NA
63	23	NA	22	23	23	21	21	21	20	NA
64	23	NA	22	23	23	21	21	21	20	NA
65	23	NA	22	23	23	21	21	21	20	NA
66	23	22	22	23	23	21	21	21	20	20
67	23	22	22	23	23	21	21	21	20	20
68	23	22	22	23	23	21	21	21	20	20
69	23	22	22	23	NA	21	21	21	20	20
70	23	22	22	23	NA	21	21	21	20	20
71	23	NA	22	23	NA	21	21	21	NA	20
72	23	NA	22	23	NA	21	21	21	NA	20
73	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
74	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
75	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
76	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
77	21	22	NA	23	23	21	NA	21	20	NA
78	21	22	22	23	23	21	21	21	20	20
79	21	22	22	23	23	21	21	21	20	20
80	21	22	22	23	23	21	21	21	20	20
81	21	22	22	23	23	21	21	21	20	20
82	21	22	22	23	23	21	21	21	20	20
83	21	22	22	23	23	21	21	21	20	20
84	21	22	22	23	23	21	21	21	20	20
85	22	22	23	23	23	21	21	21	20	20
86	22	22	23	23	23	21	21	21	20	20
87	22	22	23	23	23	22	21	21	20	20
88	22	22	23	23	23	22	21	21	20	20
89	22	22	23	23	23	22	21	21	20	20
90	22	22	23	23	23	22	21	21	20	20
91	22	22	23	23	23	22	21	21	20	20
92	22	22	23	23	23	22	21	21	20	20
93	22	22	23	23	23	22	21	21	20	20
94	22	22	23	23	23	22	21	21	20	20
95	22	22	23	23	23	22	21	21	20	20
96	22	22	23	23	23	22	21	21	20	20

Time block	Day 41	Day 42	Day 43	Day 44	Day 45	Day 46	Day 47	Day 48	Day 49	Day 50
1	20	21	20	20	19	19	20	19	19	18
2	20	21	20	20	19	19	20	19	19	18
3	20	21	20	20	19	19	20	19	19	18
4	20	21	20	20	19	19	20	19	19	18
5	20	21	20	20	19	19	20	19	19	18
6	20	21	20	20	19	19	20	19	19	18
7	20	21	20	20	19	19	20	19	19	18
8	20	21	20	20	19	19	20	19	19	18
9	20	21	20	20	19	19	20	19	19	18
10	20	21	20	20	19	19	20	19	19	18
11	20	21	20	20	19	19	20	19	19	18
12	20	21	20	20	19	19	20	19	19	18
13	20	21	20	20	19	19	20	19	19	18
14	20	21	20	20	19	19	20	19	19	18
15	20	21	20	20	19	19	20	19	19	18
16	20	21	20	20	19	19	20	19	19	18
17	20	21	20	20	19	19	20	19	19	18
18	20	21	20	20	19	19	20	19	19	18
19	20	21	20	20	19	19	20	19	19	18
20	20	21	20	20	19	19	20	19	19	18
21	20	21	20	20	19	19	20	19	19	18
22	20	21	20	20	19	19	20	19	19	18
23	20	21	20	20	19	19	20	19	19	18
24	NA	NA	NA	20	NA	NA	NA	NA	NA	NA
25	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
26	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
27	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
28	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
29	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
30	NA	20	NA	NA	NA	19	NA	19	18	NA
31	20	20	20	NA	NA	19	NA	19	18	18
32	20	20	20	19	NA	19	19	19	18	18
33	20	20	20	19	20	19	19	19	18	18
34	20	20	20	19	20	19	19	19	18	18
35	20	20	20	19	20	19	19	19	18	18
36	20	20	20	19	20	19	19	19	18	18
37	20	20	20	19	20	19	19	19	18	18
38	20	20	20	19	20	19	19	19	18	18
39	20	20	20	19	20	19	19	19	18	18
40	20	20	20	19	19	19	19	19	18	18
41	20	20	NA	NA	19	NA	19	19	18	NA
42	20	20	NA	NA	19	NA	19	19	18	NA
43	20	20	NA	NA	19	NA	19	19	18	NA
44	20	20	NA	NA	19	NA	19	19	18	NA
45	20	20	NA	NA	19	NA	19	19	18	NA
46	20	20	NA	NA	19	NA	19	19	18	NA

47	20	20	NA	NA	19	NA	19	19	18	NA
48	20	20	20	19	19	18	19	19	18	18
49	20	20	20	19	19	18	19	19	18	18
50	20	20	20	19	19	18	19	19	18	18
51	20	20	20	19	19	18	19	19	18	18
52	20	20	20	19	19	18	19	19	18	18
53	20	20	20	19	19	18	19	19	18	18
54	20	20	20	19	19	18	19	19	18	18
55	20	20	20	19	19	18	19	19	18	18
56	20	20	20	19	19	18	19	19	18	18
57	20	20	20	19	19	18	19	19	NA	18
58	20	20	20	19	19	18	19	19	NA	18
59	20	20	20	19	19	18	19	19	NA	18
60	20	20	20	19	19	18	19	19	NA	18
61	20	20	20	19	19	18	19	19	NA	18
62	20	20	20	19	19	18	19	19	18	18
63	20	20	20	19	19	18	19	19	18	18
64	20	20	20	19	19	18	19	19	18	18
65	20	20	20	19	19	18	19	19	18	18
66	20	20	20	19	19	18	19	19	18	18
67	20	20	20	19	19	18	19	19	18	18
68	20	20	20	19	19	18	19	19	18	18
69	20	20	20	19	19	18	19	19	18	18
70	20	20	20	19	19	18	19	19	18	18
71	20	NA	20	19	19	18	19	19	NA	18
72	20	NA	NA	19	NA	NA	19	19	NA	NA
73	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
74	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
75	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
76	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
77	21	NA	20	NA	19	19	NA	19	18	18
78	21	20	20	19	19	19	20	19	18	18
79	21	20	20	19	19	19	20	19	18	18
80	21	20	20	19	19	19	20	19	18	18
81	21	20	20	19	19	19	20	19	18	18
82	21	20	20	19	19	19	20	19	18	18
83	21	20	20	19	19	19	20	19	18	18
84	21	20	20	19	19	19	20	19	18	18
85	21	20	20	19	19	20	19	19	18	18
86	21	20	20	19	19	20	19	19	18	18
87	21	20	20	19	19	20	19	19	18	18
88	21	20	20	19	19	20	19	19	18	18
89	21	20	20	19	19	20	19	19	18	18
90	21	20	20	19	19	20	19	19	18	18
91	21	20	20	19	19	20	19	19	18	18
92	21	20	20	19	19	20	19	19	18	18
93	21	20	20	19	19	20	19	19	18	18
94	21	20	20	19	19	20	19	19	18	18
95	21	20	20	19	19	20	19	19	18	18
96	21	20	20	19	19	20	19	19	18	18

Time block	Day 51	Day 52	Day 53	Day 54	Day 55	Day 56	Day 57	Day 58	Day 59	Day 60
1	18	19	18	18	18	18	19	18	18	18
2	18	19	18	18	18	18	19	18	18	18
3	18	19	18	18	18	18	19	18	18	18
4	18	19	18	18	18	18	19	18	18	18
5	18	19	18	18	18	18	19	18	18	18
6	18	19	18	18	18	18	19	18	18	18
7	18	19	18	18	18	18	19	18	18	18
8	18	19	18	18	18	18	19	18	18	18
9	18	19	18	18	18	18	19	18	18	18
10	18	19	18	18	18	18	19	18	18	18
11	18	19	18	18	18	18	19	18	18	18
12	18	19	18	18	18	18	19	18	18	18
13	18	19	18	18	18	18	19	18	18	18
14	18	19	18	18	18	18	19	18	18	18
15	18	19	18	18	18	18	19	18	18	18
16	18	19	18	18	18	18	19	18	18	18
17	18	19	18	18	18	18	19	18	18	18
18	18	19	18	18	18	18	19	18	18	18
19	18	19	18	18	18	18	19	18	18	18
20	18	19	18	18	18	18	19	18	18	18
21	18	19	18	18	18	18	19	18	18	18
22	18	19	18	18	18	18	19	18	18	18
23	18	19	18	18	18	18	19	18	18	18
24	18	NA	18	NA	NA	NA	NA	NA	NA	18
25	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
26	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
27	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
28	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
29	NA	NA	NA	18	NA	18	NA	NA	NA	NA
30	18	NA	NA	18	18	18	NA	NA	18	NA
31	18	NA	18	18	18	18	18	NA	18	18
32	18	NA	18	18	18	18	18	18	18	18
33	18	18	18	18	18	18	18	18	18	18
34	18	18	18	18	18	18	18	18	18	18
35	18	18	18	18	18	18	18	18	18	18
36	18	18	18	18	18	18	18	18	18	18
37	18	18	18	18	18	18	18	18	18	18
38	18	18	18	18	18	18	18	18	18	18
39	18	18	18	18	18	18	18	18	18	18
40	18	18	18	18	18	18	18	18	18	18
41	NA	18	18	18	18	18	NA	NA	18	18
42	NA	18	NA	18	18	18	NA	NA	18	NA
43	NA	18	NA	18	18	18	NA	NA	18	NA
44	NA	18	NA	18	18	18	NA	NA	18	NA
45	NA	18	NA	18	18	18	NA	NA	18	NA
46	NA	18	NA	18	18	18	NA	NA	18	NA

47	NA	18	NA	18	18	18	NA	NA	18	NA
48	18	18	NA	18	18	18	18	19	18	NA
49	18	18	18	18	18	18	18	19	18	18
50	18	18	18	18	18	18	18	19	18	18
51	18	18	18	18	18	18	18	19	18	18
52	18	18	18	18	18	18	18	19	18	18
53	18	18	18	18	18	18	18	19	18	18
54	18	18	18	18	NA	18	18	19	18	18
55	18	18	18	18	NA	18	18	19	18	18
56	18	18	18	18	NA	18	18	19	18	18
57	18	18	18	18	NA	18	18	19	18	18
58	18	18	18	18	NA	18	18	18	18	18
59	18	18	18	18	18	18	18	18	18	18
60	18	18	18	18	18	18	18	18	18	18
61	18	18	18	18	18	18	18	18	18	18
62	18	18	18	18	18	18	18	18	18	18
63	18	18	18	18	18	18	18	18	18	18
64	18	18	18	18	18	18	18	18	18	18
65	18	18	18	18	18	18	18	18	18	18
66	18	18	18	18	18	18	18	18	18	18
67	18	18	18	18	18	18	18	18	18	18
68	18	18	18	18	18	18	18	18	18	18
69	18	18	18	18	18	18	18	18	18	18
70	18	18	18	18	18	18	18	18	18	18
71	18	18	18	18	18	NA	18	18	NA	18
72	18	NA	NA	18	NA	NA	NA	18	NA	18
73	NA	NA	NA	NA	NA	NA	NA	NA	NA	18
74	NA	NA	NA	NA	NA	NA	NA	NA	NA	18
75	NA	NA	NA	NA	NA	NA	NA	NA	NA	18
76	NA	NA	NA	NA	NA	NA	NA	NA	NA	18
77	NA	18	18	18	18	18	NA	NA	18	18
78	19	18	18	18	18	18	18	18	18	18
79	19	18	18	18	18	18	18	18	18	18
80	19	18	18	18	18	18	18	18	18	18
81	19	18	18	18	18	18	18	18	18	18
82	19	18	18	18	18	18	18	18	18	18
83	19	18	18	18	18	18	18	18	18	18
84	19	18	18	18	18	18	18	18	18	18
85	19	18	18	18	18	18	18	18	18	18
86	19	18	18	18	18	18	18	18	18	18
87	19	18	18	18	18	19	18	18	18	18
88	19	18	18	18	18	19	18	18	18	18
89	19	18	18	18	18	19	18	18	18	18
90	19	18	18	18	18	19	18	18	18	18
91	19	18	18	18	18	19	18	18	18	18
92	19	18	18	18	18	19	18	18	18	18
93	19	18	18	18	18	19	18	18	18	18
94	19	18	18	18	18	19	18	18	18	18
95	19	18	18	18	18	19	18	18	18	18
96	19	18	18	18	18	19	18	18	18	18