ENTS 689 Project 1

Few steps to be followed for both Online and Offline algorithms.

1. Use File I/O of C programming to read the contents of file using a while loop and fscanf to delimit the file and save the contents of the file in a structure array like.

   Struct data
   {
   int pid, int time, int start_elevator, int end_limiter, int diff, int waitingtime;
   }d[];

   The above structure array stores data from the file.
2. From the command line input, create a number of arrays which corresponds to the number of lifts with an array size of 10 each.
   Here, the number 10 corresponds to the maximum no of passengers accommodated in the list.

Online:

For Online scheduling,

1. All the lifts start at position of Floor 1.
2. Simulation starts at time t=0
   While(t<86400) the max time is 1 day which is 24*60*60.
   While starting with the simulation, all the elevators are in zeroth floor.
   Until the first passenger arrives let say he arrives at the 10^{th} floor, as per my logic the elevator starts moving only on the arrival of the first passenger.

   Let say, the passenger arrives at 10^{th} floor at 10 sec.

   Moving the lift-up and down:
   Write function for moving the elevator from one floor to another. The function should get the list of all elevators along with a Boolean set for each elevator which tells whether you need to move the elevator up or down.

   Two strutures to handle the elevators

   Structure elevators{ int floor, bool moveup, bool movedown, bool first floor, time}
   Struture first_elevator{int passenger_id, int arrival_time, start_floor, end_floor, time to reach destination}

   While moving the elevator, change the variables in both the structures.

   As per the question, the elevator when a new passenger enters and wants to move up it takes 3 seconds to go up the floor afterwards 2s. I would like to maintain a Boolean element

which tells whether the elevators stops in the last floor or not. This helps us to handle the above-mentioned condition completely.

3. Let me help understand my algorithm with various scenarios, based on the input file provided.

   Simulation starts at time t=0
   Until the new passenger came in, the elevator will never move it stays where it already was.

   At time t=10s, the passenger 1 comes at floor 4. At the time t=10s the elevator 1 starts moving up as per the function I have written to move the elevator move. For the elevator one, you are changing the bool moveup as true and bool firstfloor as true and time as 3. After 3 seconds at t=13s, the elevator moves to the first floor, there reverse the bool firstfloor and then call the same function – During this particular period of moving elevator 1, Other elevator 2 is not moving in this scenario. To be clear, I have assigned the first passenger to the elevator one.

   For each and every passenger, if a particular elevator stops at the particular location the passenger gets appended to an array pertaining to the elevator based on the movement of the elevator. I will check other elevators which are nearer to the passenger and based on its current movement direct the elevator to the destination.

   For each and every second, the elevator moving function gets executed.

   At t=17s, a new passenger shows up at floor 2. During this time, the elevator 1 was already in floor 3 so it missed the passenger 2. So, I am moving the elevator 2 to take care of passenger 2.

   At time t=19s, the elevator 1 reaches the floor 4 to take in the passenger 1, by t=19s it takes the passenger in and is moving up.

   As explained for the above scenario, the elevator is moved. At t=22, the elevator reaches floor 2, there it stops and it takes 2 seconds to take a passenger into the elevator and it get closed, once the passenger enters the elevator, set the firstfloor Boolean up.

   I am maintaining two kinds of structure for maintaining the elevator. One for holding the passengers like pushing in and out. Other for maintaining the state of the elevator – current floor, movingup or movingdown those kinds of states.

   At time t =33s, the elevator reaches floor 15 it drops the passenger 1 down. Once the elevator is empty without any passenger, it stays where it is. Once the new passenger shows up, then only the elevator which needs to reach the passenger moves up. In this case, there are scenarios where the wait-time of the passenger may increase. Here, there is a trade-off between the wait time and the no. of floors travelled by the elevator. I am trying to optimize both of them.

When a passenger enters into the elevator, append the passenger into the elevator array and execute the same code for all the elevators.

I have 2 ways to handle it.

1. Keep the elevator at the same position
2. Move the elevator to the middle floor if it is 100 floors(Move it to 50th floor).

Here, to handle all the other criteria we need to do check the length of the elevator. I new passenger comes in, I will check the size of the closest elevator if it is less than 10 and the direction of the elevator matches with that of the customer, I will take him in the elevator else I had to move other elevator towards the spot.

Since, I cannot satisfy the criteria of both No.of floors travelled and Waiting time of the passengers. I will try to optimize the number of floors travelled.

During each and every second of the simulation, I will add the contents of the Elevator structure to s_e.txt and schedule.txt. After successful dropping of passenger to the particular location, I will append the data to s_p.txt.

Append the data for the each and every passenger into the elevator array. When the passenger gets out of the elevator, remove the passenger from the array corresponding to the elevator and append the data into the s_p.txt.

For calculating the waiting time, in the structure calculate the waiting time by incrementing 1s for each iteration till the passenger enters the elevator. To find the max value, use the normal algorithm to find the max value.

To calculate the number of floors travelled by each elevator, increment the array elements for each and every iterations. This value is incremented each and every time. I will try to optimize the number of floors to be travelled by the elevators.

Offline algorithm:

In case of offline algorithm, I will use the same data structure which consists of structure which will contain all the contents of the file.

I will try to reuse almost all the code of Online function for Offline with few exception like making decision while knowing all the possibilities for the day.

Following functions will be the same for both online and offline,

1. The data structure used for storing contents of the file.
2. The structure which holds the state of elevators at each and every seconds.
3. The function which allows the movement of the elevators are the same.
4. The function which will add contents to the output file are the same.

The assumption I had for the offline implementation is that I would like to move the elevators up from Floor 1 with the interval of 10s for each and every elevator.

I have few problems with respect to the implementations.

1. I have a doubt here, if we have 2 elevators, Can I allow elevator 1 to go from 0 towards the max no. of floors and elevator 2 to go from max to 0.
2. What is the threshold for the number of seconds to wait for making the decision.

   In case of offline the main assumption I am taking is that the elevator will be moving continuously without stopping allowing it to make decision.

   Let us consider a scenario, at floor 10 a passenger comes at 300 seconds, the elevator with 5 passenger and the elevator is at floor 10 at 290s, I would wait for the remaining seconds. The threshold I would like to keep for this is 10s which will handle the problem but this will add to the overall waiting time.

   Consider a case when the elevator is empty and moving-up and a new passenger coming at the floor below at the time of 30s then change the direction of movement that allows the elevator to reach the particular floor.

   I can allow an elevator to change its direction of motion only when it is empty, if not empty the elevator continue to move the same way. Based on the position and time taken to go and get into the position, move the elevator accordingly.

   Calculation of the waiting time, no. of floors travelled by a lift and handling the overflow are same as that of Online.