Nothing brings people together like food...

Restaurant

Ahmed Ibrahim El-Garf 1180262
Nouran Ahmed Mohamed 1180043
Youssef Mokhtar Mohamed 1180054
Hannah Hatem Abdel-Hakeim 1180273

VIP Orders : Priority Queue

There is a distinct priority for each VIP order based on four factors. This priority decides which order will be served and finished first. So we came up with a function that calculates the priority of VIP orders according to the four factors and gave them reasonable weights from our point of view and we will discuss them as follows:

Priority = (4xTotal Money)+(3xOrder Size)+(2xDistance)+(1/Arrival Time)-(ID/1000000.00).

It's too obvious that money has the highest weight giving consideration that the number one target for a restaurant owner is the profit so orders with more money should be the most ones to be taken care of and served well and fast because they paid more so they wait for special treatment.

We gave the second highest rate to the order size as the more the order size gets bigger the more it takes time to finish and serve it in a good way to the customer so we gave it the second rank to be notified that they must be assigned quickly to a cook because it takes more time.

Finally we come to the distance that is restricted only on delivery orders and it has a higher priority than arrival time because the more distance increases the more it takes time to get it to the customer so we gave it a higher priority but if the order isn't a delivery one so we assume that distance equals zero and then arrival time will be the third one in the priority rank.

We made this function to sort the orders according to the priority of each one of them so it's not necessary that the first order to come in is the first order to be served and come out so we chose Priority Queue to achieve this sorting correctly and not in the traditional way (FIFO).

If two or more orders were equal in priority then we take into consideration the ID number.

For clarity, our priority equation will be used in phase 2 so, we commented this part and made a temporary priority depend on arrival time and ID only for phase 1.

Vegan Orders : Queue

Vegan orders are much simpler than VIP as the first order that comes in is the first order to be served and come out (FIFO) so it's arranged according to arrival time so we chose Queue to achieve this as it's the most convenient data structure.

Normal Orders : Queue

Same goes for normal orders as it's arranged also according to arrival time so the first order that comes in is the first order to be served and come out (FIFO). But they have some special operations which are cancellation and promotion and these operations are for normal orders only. So we made some functions to access the orders to be cancelled and delete it from the queue and access the orders to be promoted and transfer them to VIP Orders Priority Queue.

Cooks: Linked Lists

All cooks are implemented using Linked Lists. There are three Linked List one for each type VIP, Vegan and Normal. The VIP especially will be sorted according to speed, as required in bonus section the ones with higher speeds are the first to serve orders. Implementing cooks using linked lists made it easier to traverse through each list which eased up the operation of who has to go for a break at the current time step and to keep track of who's injured and sort according all the previously mentioned points. One of the major operations performed on cooks is assigning orders to them and moving them away from available lists then retrieving them back after finishing the order, then resorting them according to speed and that couldn't have been easier by choosing Linked List as the data structure because it doesn't limit us with size or shifting if I inserted something in the middle.

“In EVENTS we use queues to store the data from the input file”

Arrival Event:

This event is responsible for receiving the new orders , adding them in queue and storing their information (type , ID , time step , money, size)

Promotion Event:

This is valid only for NORMAL TYPE , we can get extra money and promote the order to be VIP type them remove the order from the NORMAL QUEUE and add it to VIP QUEUE and re-arrange them according to the priority equation.

Cancellation Event:

We can cancel the order as long as it is not assigned to a cook and it must be NORMAL we used the queue that we stored the orders in , then we compare the given ID with the orders’ ID in the queue of orders….as it is a queue of pointers to order so we need to convert the queue to array by function (ToArray) to get the pointer then use(-> GetID()) , we will have the pointer that refers to the order we want to cancel. If we want to cancel you may call Function: cancelOrder().