



Faculty of Engineering  
Cairo University  
Computer Engineering Department

# **CMP102**

## **Data Structures and Algorithms Project**

### **Restaurant**

### **MAIN STREAM**

**Spring 2020**

**Team :**

Name	Section	BN
Weaam Bassem Hosney	2	35
Yousef Gamal Reda	2	38
Yousef Aatef Abdo	2	40
Yousef Mustafa El-Mahdy	2	41

## 1. Events

**Data Structure:** Queue

**Explanation :** Queue has a very useful feature which is FIFO (First In First Out) ,and because the events are orders which can be arrived canceled or promoted ,Then the logic of Serving is that the first one to arrive is the first one to be served , as also in the input file the arrival time is sorted in ascending order so the queue will be so suitable in logic of serving and Complexity ,it will be almost constant in enqueueing and dequeuing  $O(1)$  .

## 2. Cooks

**Data Structure:** List ADT

**Explanation:** We Spilt the Cooks into 4 List ADTs Unavailable cooks , VIP, Vegan, and Normal cooks lists and the reason for using List ADTs in them is that we may need to get a cook from the middle of the list not only to peek the front of the list so we did not use Queue Data Structure in Cooks, Also the Unavailable cooks may be unavailable for many reasons such as they are having a break or they are finishing orders which are different in their size and cooks are different in their speed .

## 3. Orders

### i. Vegan Orders

**Data Structure:** Queue

**Explanation:** Because the mechanism of serving them is the First order to be in the queue is the first order to be served and get dequeued (FIFO).

## ii. **VIP Orders**

**Data Structure:** Priority Queue

**Explanation:** Like any mechanism of Orders serving that First In First Out So they should be Queue but they have another feature which is the priority, as VIP orders have priority equation which depends on Order Arrival Time, Order Money, and Order Size.

We used the following priority equation in our code

Equation =  $MONEY + (1000 / (\text{float})TS) + SIZE$ ;

So the suitable Data Structure is priority queue.

## iii. **Normal, In\_Service and Finished Orders**

**Data Structure:** List ADT

**Explanation:** As we may need to take an order from the middle for example in Normal orders for cancelation and promotion events, As it is not necessary that the order which will be promoted or cancel is on the front of the list so we cannot use Queue data structure , but we can use list which we can get the order from the front and add an order from the tail of the list and also extract an order from the list which is not available for queue data structure.