|  |  |
| --- | --- |
| Cairo University, Faculty of Engineering  Computer Engineering Department  Data Structures and Algorithms | Spring 2022 |

**Data Structure and Algorithms**

**Project Phase 1 Report**

**Team Name:**  4-6pm T07

**Team Email:** mariamgamal70@gmail.com

**Number of Members:** 4

**Members’ Info:**

|  |  |  |
| --- | --- | --- |
| **Members Name** | **ID** | **Email** |
| Mariam Gamal Tawfic | 1200335 | Mariamgamal70@gmail.com |
| Malak Akram Mohammad | 1200914 | malak.ali021@eng-st.cu.edu.eg |
| Sherif Ehab Sayed | 1200958 | Sharif.Mohammad02@eng-st.cu.edu.eg |
| Yara Ihab Mohammad | 1200452 | Yara.ihab02@gmail.com |

**Project Data Structures:**

|  |  |  |
| --- | --- | --- |
| **List Name** | **Chosen DS** | **Justification** |
| Cargo list (Waiting to be loaded):  NormalCargos  SpecialCargos  VIPCargos | Linked list for normal  Queue for special  Priority Queue for VIP | Linked list-> to be able to cancel and promote cargo  Queue-> FIFO Priority queue-> due to priority equation  Enqueue and dequeue O(n) |
| Loading Cargos:  LoadingCargos | Priority Queue inside truck class | Priority to those closest to the truck’s tailgate.  Queued in order of delivery distance. (Nearest delivery distance is dequeued first)  Enqueue and dequeue O(n) |
| Delivered Cargos:  DeliveredCargos | Queue for each type | It’s a queue because they’re being dequeued after they have been delivered for a certain amount of time.  For each type to make it easier to print like program interface  No traversal required  Enqueue complexity O(1) |
| Truck list (available empty truck list)  NormalTruckQueue  SpecialTruckQueue  VIPTruckQueue | Queue for each type | So that assignment of each cargo type is matched with its available truck type  Enqueue and dequeue O(1) |
| Available loading trucks: LoadingNormalTrucks  LoadingSpecialTrucks  LoadingVIPTrucks | Queue for each type | First truck loaded is the first truck to be moved (FIFO).  Enqueued in arrival order.  Enqueue and dequeue O(1) |
| Moving trucks list:  MovingTrucks | 1 Priority queue for all types | Based on FIFO  First move first arrive.  The least delivery time required is the first out.  Enqueue and dequeue O(n) |
| Trucks under checkup list  (Not available trucks)  NormalTrucksUnderCheckup  SpecialTrucksUnderCheckup;  VIPTrucksUnderCheckup; | Priority Queue for each list | Based on FIFO for each type.  First done checkup, first out.  For each type as each type has its own checkup time.  Enqueue and dequeue O(n) |
| Events list:  Eventlist | 1 Queue for all events according to time | Based on FIFO  Enqueue and dequeue O(1) |

|  |  |  |
| --- | --- | --- |
| Linked List Complexities | Linked Queue Complexities | Priority Queue Complexities |
| - DeleteAll, InsertEnd, printList, DeleteSpecificNode, DeleteEnd, getLastNode : O(n)  Note: All other functions’ complexity is O(1) | PrintQueue, ~Linkedqueue, copy constructor: O(n)  Note: All other functions’ complexity is O(1) | Enqueue Ascending,  Enqueue Descending,  PrintList, getLastNode: O(n)  Note: All other functions’ complexity is O(1) |