Cairo University, Faculty of Engineering
Computer Engineering Department
Data Structures and Algorithms
CMPN1020

Spring 2021

Data Structures and Algorithms
Project Phase 1 Report

Team Name: Mars Astromission
Team E-mail: omarkh200165@gmail.com
Number of members: 4

Members' information:

Member Name	ID	E-mail
Omar Khaled Ali	9202938	omarkh200165@gmail.com
Nour Ziad Almulhem	9204005	nouralmulhem@gmail.com
Kareem Ashraf Mohammed	9203063	Kemokhalifa5@gmail.com
Mahmoud Abdelhamid ali	9203420	mahmoud13abdelhamid@gmail.com

Project Data Structures:

List Name	Chosen DS	Justification
Waiting Mountainous Mission List	Queue	**Operations:** Enqueue () O(1) Dequeue() O(1) IsEmpty() O(1) IsFull() O(1) Peek() O(1) Mountainous missions should be assigned based on a first-come first-served basis, they are the only missions could be promoted or cancelled thus we put it in a distinct list to be easier to find the mission, we need a function to search for the mission ID or mission WD and remove it from the current list then add it to another one or delete it complexity of the function will be O(n).

Waiting Emergency and Polar Mission List	Pri-Queue	**Operations:** Enqueue () O(n) Dequeue() O(1) IsEmpty() O(1) IsFull() O(1) Peek() O(1) Emergency and Polar Missions should be ordered based on highest priority first-served basis. this list will be sorted due to the type of mission first (Emergency have a higher priority) and sorted with a priority equation for deciding which of the available emergency missions should be assigned first. Emergency missions with a higher priority are the ones to be assigned first. then Polar Mission with constant negative Priority That makes first in first out but after all Emergency mission be executed.
Missions in Execution	Pri-Queue	**Operations:** Enqueue () O(n) Dequeue() O(1) IsEmpty() O(1) IsFull() O(1) Peek() O(1) In-Execution missions must be sorted in Ascending order due to their CD the priority of the Enqueue function will be 1/CD.
Completed Missions	Queue	**Operations:** Enqueue () O(1) Dequeue() O(1) IsEmpty() O(1) IsFull() O(1) Peek() O(1) The output file must be sorted in Ascending order due to CD which is already done in in-Execution mission list.

Dummy list	Queue	**Operations:** Enqueue () O(1) Dequeue() O(1) IsEmpty() O(1) IsFull() O(1) Peek() O(1) This list is mainly to help us in Promoting OR Canceling an mission and getting the mission with certain ID from any position in Waiting Mountainous Mission List
Temporary list	Pri-Queue	**Operations:** Enqueue () O(n) Dequeue() O(1) IsEmpty() O(1) IsFull() O(1) Peek() O(1) The output lines must be sorted by CD in ascending order. If more than one mission is completed on the same day, they should be ordered by ED.
Events list	Queue	**Operations:** Enqueue () O(1) Dequeue() O(1) IsEmpty() O(1) IsFull() O(1) Peek() O(1) the input lines of all events are sorted by the event day in ascending order.
Available Emergency Rovers list	Pri-Queue	**Operations:** Enqueue () O(n) Dequeue() O(1) IsEmpty() O(1) IsFull() O(1) Peek() O(1) We have chosen this data structure to give the priority for the higher speed rovers to execute missions first. We split it into 3lists according to its type because some missions can be executed with only some kinds of Rovers.

Available Mountainous Rovers list	Pri-Queue	**Operations:** Enqueue () O(n) Dequeue() O(1) IsEmpty() O(1) IsFull() O(1) Peek() O(1) We have chosen this data structure to give the priority for the higher speed rovers to execute missions first. We split it into 3lists according to its type because some missions can be executed with only some kinds of Rovers.
Available Polar Rovers list	Pri-Queue	**Operations:** Enqueue () O(n) Dequeue() O(1) IsEmpty() O(1) IsFull() O(1) Peek() O(1) We have chosen this data structure to give the priority for the higher speed rovers to execute missions first. We split it into 3lists according to its type because some missions can be executed with only some kinds of Rovers.
Checkup list	Pri-Queue	**Operations:** Enqueue () O(n) Dequeue() O(1) IsEmpty() O(1) IsFull() O(1) Peek() O(1) This data structure is used to give the priority for the lower checkup-hours rovers to come first and be ready again for executing the missions.
In-execution Rovers list	Pri-Queue	**Operations:** Enqueue () O(n) Dequeue() O(1) IsEmpty() O(1) IsFull() O(1) Peek() O(1) Here this data structure gives the priority for the rovers which finish the execution first to return to the available list again for the new missions.