



Cairo University
Faculty of Engineering



DATA STRUCTURES AND ALGORITHMS

MARS EXPLORATION PROJECT

PHASE1

Members name	ID	Email
Abdelrahman Mohamed Ezzat	1190158	rahmanezzatz14@gmail.com
Omar Mohamed Ahmed Amin	1190204	oamin.omar@gmail.com
Amr Ahmed Abdel Zaher	1190074	amr23034@gmail.com
Yomna Osama Hussien	1190203	yomna.osamma@gmail.com

Project Data Structures

List Name	Chosen DS	Justification
<i>Waiting of mountainous missions</i>	Hash table + Queue	<ul style="list-style-type: none"> As the ID of each mission is unique so hash table is used to find the mission to cancel and promote easily, to be performed in $O(1)$, and using the queue to keep the order of insertion. Operations: find $O(1)$, delete $O(1)$, move to execution $O(n)$.
<i>Waiting of polar missions</i>	Queue	<ul style="list-style-type: none"> As there are not any special operations on polar missions, so missions are executed according to the order of insertion. Operations: move to execution $O(n)$.
<i>Waiting of emergency missions</i>	Priority Queue	<ul style="list-style-type: none"> The missions are sorted according to the calculated equation (priority equation). Operations: move to execution $O(n)$.

<i>In execution missions list</i>	Priority Queue and if we did the bonus, we will add a Hash table	<ul style="list-style-type: none"> The missions are sorted in an ascending order according to the Completion-day of the mission so that they will be sorted in the order of completion when moved to the completed mission list. Operations: move to completion $O(n)$. Bonus, we will use hash table in order to search for the failed mission easily with complexity $O(1)$.
<i>Completed missions list</i>	Queue + Priority Queue	<ul style="list-style-type: none"> The queue is sorted according to the completion day from dequeuing the execution list. In case of same completion day, a priority queue is used before inserting in the queue to be sorted in order of event day to be sorted for the output file. Operations: write to the output file.
<i>Events list</i>	Queue	<ul style="list-style-type: none"> As it read from the input file in order of the event day. Operations: execute the appropriate event $O(n)$.
<i>Array of available rovers</i>	Array of priority queue and another 3 priority queues for the maintenance(bonus).	<ul style="list-style-type: none"> The array contains 3 priority queues in order to sort the rovers according to their specific speed for the bonus and for the maintenance rovers in order to sort them according to their maintenance duration. Operations: move to in mission $O(n)$.

<i>Rovers in mission</i>	Priority queue	<ul style="list-style-type: none"> • As it is arranged according to the lowest completion day of the mission. • Operations: move to available or to checkup $O(n)$.
<i>Rovers in checkup</i>	Priority queue	<ul style="list-style-type: none"> • As it is sorted according to the rovers having the least checkup day, where any rover in this list is unavailable unlike the maintenance rovers in the bonus. • Operations: move to available $O(n)$.

Output function

Statistics

This function counts each type of missions per day with complexity $O(n)$.