Data Structures and Algorithms Project
Phase1 Report

Team Name: T 08 **Number of members: 3**
Team Email: s-mo_ayman@zewail.city.edu.eg

Members Info:

Name	ID	Email
Muhammad Ayman	201801356	s-mo_ayman@zewail.city.edu.eg
Hajar Fawzi	201900537	s-hajar.alshafai@zewailcity.edu.eg
Riham Ahmed Mohamed	201901639	s-riham.mohamed@zewailcity.edu.eg

List Name	Chosen DS	Justification
Fighters	**Priority Queue**	We chose priority queues so that it would sort the fighter in the same way the castle would attack them (same as the enemy picking criteria) We thought BST and sort items by their arrival time but the tree wouldn't balance (would turn into a linked list) and the complexity of insertion / deletion / search would be the same as linked lists. Note that in this case heaps and priority queues can equally fit here but we chose to work with priority queues (implemented using heaps)
Healers	**Stack**	Since the castle would attack the fresh healers who had just joined the forces. (Last-Come-First-Serve)
Freezers	**Queue**	Since the castle would attack the freezers who arrived firstly.(First-Come-First-Serve)

Inactive	**Queue**	Since they are loaded form the input file in an ascending order (according to their arrival time) which is the same order it will be activated in. (First-Come-First-Serve)

Note:
The frosted enemies are not stored in a separate list but known as a property of the enemies. Whenever an enemy gets frosted its property “IsFrosted” is turned into Ture and the frosted enemies count increases.

1- Do you need a separate list for each enemy type? Why?
Yes, for active only!
As every enemy type has its way to be stored and attacked by Castle.
First for fighters, We needed a **Priority Queue**!
The castle should pick fighters depending on their distance, power, and health, status (active or frosted), remaining time steps for an enemy to end the reload period.
Then for Healers,We need a **Stack !**
As a fresh healer who has just joined the forces is picked first. This means that healers are picked in the reverse order of their arrival (Last-Come-First-Serve).
Finally for Freezers, We needed a **Queue**!
As a freezer that arrived first should be attacked first (First-Come-First-Serve)

2- Do you need to store killed enemies? In separate lists or in one list? When should you get rid of them to save memory?
No, we will not save them. Since we can save each enemy data and statistics just when it is killed to the output file automatically. (that would save space but that might not be possible since the project document states that we have to mission whether the game is win or not in the first line)
If we can just move the game status line to the last line in the output file, then we can do the aforementioned.
If not then we will have to save them in a queue since they will be written in the file in an ascending order according to their killed time step.