CS 2341 – Assignment 2

Objective:

The objective of this assignment is to understand and implement priority queues in a real-world scenario where tasks need to be scheduled and processed based on their processing time and priority. You will develop a system to manage and prioritize a large number of tasks efficiently using a priority queue.

Scenario:

You are developing a task scheduling system for a large cloud computing platform. The platform receives a continuous stream of tasks, each with different processing time and priority level.

Task 1:

You are given N jobs, where each job j requires tj seconds of processing time. You need to complete all of the jobs but want to maximize customer satisfaction by minimizing the *average completion time* of the jobs. Completion time of a job is defined as the total time elapsed from the start of the schedule until the job is completed.

Using the Shortest Processing Time First (SPT) rule, you need to design a job scheduler that:

1. Schedules jobs in increasing order of processing time.
2. Outputs the order in which the jobs should be executed.
3. Calculates the average completion time of all jobs.

To solve this problem, you will implement the scheduler using a priority queue (min-heap), where the jobs are prioritized based on their processing times.

Input: Job Id, Processing time
1 4
2 2
3 8
4 6
5 1

Output:
Execution order: [5, 2, 1, 4, 3]
Average completion time: 9

Task 2: Modify the scheduler to handle *priority classes,* where each job has a priority level and jobs with higher priority (e.g., priority level 1 is higher than priority level 2) should be scheduled before jobs with lower priority. Jobs within the same priority class should still follow the SPT rule.

```
Input: Job Id, Processing time, Priority class
1 4 1
2 2 2
3 8 2
4 6 1
5 1 2
```

```
Output:
Execution order: [1, 4, 5, 2, 3]
Average completion time: 11.8
```

Task 3: Simulate an environment where jobs arrive dynamically (i.e., jobs arrive at different times), and you need to handle these dynamically using a priority queue. Use SPT rule for prioritizing jobs.

```
Input: Job Id, Processing time, Arrival time
1, 4, 0   # Job 1 arrives at time 0, takes 4 seconds
2, 2, 1   # Job 2 arrives at time 1, takes 2 seconds
3, 8, 2   # Job 3 arrives at time 2, takes 8 seconds
4, 6, 3   # Job 4 arrives at time 3, takes 6 seconds
5, 1, 5   # Job 5 arrives at time 5, takes 1 second
```

```
Output:
Execution order: [1, 2, 5, 4, 3]
Average completion time: 10.2
```

Explanation:

At time 0: Job 1 arrives and is executed first (completion time = 4 seconds).
At time 1: Job 2 arrives, but Job 1 is still running.
At time 4: Job 1 finishes, and Job 2 starts (completion time = 6 seconds).
At time 5: Job 5 arrives while Job 2 is still running.
At time 6: Job 2 finishes, and Job 5 starts (completion time = 7 seconds).
At time 7: Job 5 finishes, and Job 4 starts (completion time = 13 seconds).
At time 13: Job 4 finishes, and Job 3 starts (completion time = 21 seconds).

Additional instructions:

There is an input file for each task. Use those files as your input stream. i.e. task1-input.txt

Deliverables:

All results, concatenated into a text file
Source file(s)

All files should be uploaded to Github. You will submit the link to your GitHub repository on Canvas. You can work in pairs for this assignment. Team members' names and IDs should be included in a README file.