SELF PROJECT- Job Scheduler using Priority Queue(in C++)

Main topics used: Heap(Priority Queue),Greedy Algorithms

Objective:

To implement a job sequencing algorithm that schedules jobs with individual deadlines and profits in such a way that:

- **Each job takes exactly 1 unit of time.**
- **No two jobs overlap.**
- **The total profit is maximized.**

Problem Description:

Given a list of n jobs, where each job has:

- **A job ID (like 'A', 'B', 'C'...)**
- **A deadline (unit of time before which it should be completed)**
- **A profit (earned only if the job is completed on time)**

You must schedule jobs to maximize total profit using:

- **Greedy strategy**
- **Priority Queue logic (here, simulated by sorting in descending profit)**

CODE:

```
#include <iostream>
#include <vector>
#include <queue>
#include <algorithm>
using namespace std;

struct Job {
```

```
    char id;
    int deadline;
    int profit;
};

// Custom comparator for sorting jobs in decreasing order of profit
bool compare(Job a, Job b) {
    return a.profit > b.profit;
}

int main() {
    // Sample list of jobs {id, deadline, profit}
    vector<Job> jobs = {
        {'A', 2, 100},
        {'B', 1, 19},
        {'C', 2, 27},
        {'D', 1, 25},
        {'E', 3, 15}
    };

    // Sort jobs by descending profit
    sort(jobs.begin(), jobs.end(), compare);

    // Find maximum deadline to determine number of time slots
    int maxDeadline = 0;
    for (auto job : jobs)
```

```
    maxDeadline = max(maxDeadline, job.deadline);

// Initialize time slots as free (-1)
vector<char> schedule(maxDeadline, '-');

int totalProfit = 0;

// Try to schedule each job
for (auto job : jobs) {
    // Find a free slot from job.deadline - 1 to 0
    for (int t = job.deadline - 1; t >= 0; t--) {
        if (schedule[t] == '-') {
            schedule[t] = job.id;
            totalProfit += job.profit;
            break;
        }
    }
}

// Output the scheduled jobs
cout << "Scheduled Jobs: ";
for (char slot : schedule)
    cout << (slot == '-' ? "[empty] " : string(1, slot) + " ");

cout << "\nTotal Profit: " << totalProfit << endl;
    return 0;    }
```