PRACTICAL ASSIGNMENT
DATA STRUCTURE APPLICATION

Assignment Description

In a **job scheduling problem**, the objective is to schedule of tasks to complete the job using the minimum amount of time or processors. Planning and implementing a proper schedule to satisfy a set of constraints is fundamental to many application. A critical aspect of any parallel processing system is the algorithm mapping tasks to processors. Poor scheduling can leave most of the expensive machine sitting idle while one bottleneck task is performed. Assigning people to jobs, meetings to rooms, or courses to final exam periods are all different examples of scheduling problems. Scheduling problems differ widely in the nature of the constraints that must be satisfied and the type of schedule desired.

Example: Assigning people to jobs with dateline

Given N jobs where every job is represented by following elements:

1. Start time
2. Finish time
3. Profit or value associated
4. Deadlines

Let us consider, a set of n given jobs which are associated with deadlines and profit is earned, if a job is completed by its deadline. These jobs need to be ordered in such a way that there is maximum profit. It may happen that all of the given jobs may not be completed within their deadlines.

Assume, deadline of $\mathbf{i^{th}}$ job J_i is d_i and the profit received from this job is p_i. Hence, the optimal solution of this algorithm is a feasible solution with maximum profit.

Propose suitable data structures to store the information of jobs completed by its dateline, and construct solutions in minimum TWO(2) different strategies (data structure or any other optimal algorithms) for the job scheduling problem. Compare the number of job completed within the dateline by the strategies on randomly generated instances.

Further Information

Prepare the necessary data file(s) for building the data structure(s) needed in your application. You may give additional assumptions for your application.

To make your program more robust and avoid problems at run time, do as much status/error checking as you could in your program. And, good organization of the code and meaningful variable names would help readability, and liberal use of comments can help the marker understands what the program does and why.

In addition, provide UML diagram(s) to illustrate the design of your program and provide a flowchart for job scheduling algorithms.

Assessment and Submission

This is a group assignment. Form a group of 2 or 3 members, preferably from same programme as yours. Prepare a report (preferably using word processing software) to answer the questions given above.

Your **REPORT SHOULD CONTAIN** the following:

1. proposed solution – data structures and strategies for the job scheduling problem
2. flowchart for job scheduling algorithms
3. UML diagrams for illustrating the design of Java program
4. print out of the Java program
5. sample of input data and test cases
6. sample output(s) of your program
7. discussion/complexity analysis of the application
8. soft copy of the Java program (.java, .class and .bat files and upload to WBLE)

Do remember to print the assignment marking sheet and attach as the FIRST PAGE of your report.

The total mark of this practical assessment is 100. The 100 marks will contribute 20% of your final mark. The report will be marked for *correctness*, *completeness*, *presentation style*, and *relevant use of diagrams/tables/graphs*, etc. And the program implementation will be marked for *correctness*, *completeness*, *programming style*, *adequate testing* and *documentation/comments*. It's your responsibility to understand the requirements of the tasks and prepare well for your submission. You might be asked questions about the works you submit to ensure that you understand them.

Plagiarism

It is important that your solutions to the practical assignment be your own work. It is perfectly acceptable to seek help and advice when completing the practical assignment, but this must not be taken to the point where what is submitted is in part someone else's work.