

Programming Assignment 1

Total Marks: 10

Deadline: March 5, 2023

Task

The following task is a variant of a job scheduling problem.

Consider a set of m jobs which needs to be scheduled on n days. Each job takes exactly one day to complete and on a given day at most one job can be scheduled. Each job j , has a release time r_j and a deadline d_j , where r_j and d_j are two days with $r_j \leq d_j$. The job j can be scheduled on a day k only if $r_j \leq k \leq d_j$. You have to pay a per day cost for a machine that is required for the jobs. Interestingly, the per day cost for the machine keeps fluctuating depending on the global demand. Let us say we know the cost c_k (a natural number), associated with every day $1 \leq k \leq n$. The task is to find a schedule for all jobs such that the total cost of the allocated days is minimized. Or when it is not possible to schedule all the jobs, then say so. Some kind of greedy ideas should work to find out if scheduling is possible and to find an optimal schedule.

Example 1:

Job	Release time	Deadline	Day	Cost
1	1	2	1	30
2	2	3	2	10
			3	20

Possible schedule 1: Job 1 on day 1; Job 2 on day 2. **Total cost:** 40

Possible schedule 2: Job 1 on day 2; Job 2 on day 3. **Total cost:** 30

Possible schedule 3: Job 1 on day 1; Job 2 on day 3. **Total cost:** 50

So, schedule 2 is the best in terms of cost.

Example 2:

Job	Release time	Deadline	Day	Cost
1	2	4	1	30
2	1	5	2	10
3	2	3	3	20
4	3	4	4	10
5	2	3	5	40

In this example, it is not possible to schedule all the jobs. This is because jobs 1, 3, 4, and 5 together only have total 3 days available.

Instructions

Input contains $2 + m + n$ lines.

Line 1: m (the number of jobs)

Line 2: n (the total number of days)

Line $2 + \ell$ for $1 \leq \ell \leq m$: $r_\ell d_\ell$ (release time and deadline for ℓ th job)

Line $2 + m + k$ for $1 \leq k \leq n$: c_k (the cost of k th day)

Output must contain 2 lines:

Line 1: 0/1 (0 if it is not possible to schedule all the jobs, and 1 otherwise)

Line 2: The minimum possible total cost (if no schedule is possible, this can be an arbitrary number)

- Programming Language: C++. We will compile your code with g++. Make sure that it works.
- Submission: put your code in a file named XXX.cpp where XXX is your roll number. Also, write a short explanation (a paragraph) of what your algorithm does, put this in XXX.pdf. The two files should be uploaded on Moodle (do not zip/compress).
- Given files: In the **GreedyAssignment** folder, you will find: (i) helper.cpp (a c++ code showing expected input/output, feel free to use) (ii) A few sample input and output files.
- Running time: we will test your code on some similar size instances as given in the sample input files. There are some small size inputs and a large size input in the sample files. For the large size test case, your code should take only a few seconds. This will work, if your running time is $O(n^3)$ or smaller.
- Academic integrity: Mention all references if you have referred to any resources while working on this assignment in the pdf. You are supposed to do the assignment on your own and not discuss with anyone else. We will do a plagiarism check on your submission using MOSS. It's fairly sophisticated and can detect even when you have made modifications in someone else's code. Any cases found with significant overlap will be sent to DADAC. If DADAC finds it to be a case of plagiarism, then the penalty is zero in the assignment and final course grade reduced by 1 point.
- Grading: Your code will be tested on a set of test cases which is a superset of the ones provided. The test cases will be of varying sizes. Your marks will be proportional to the number of test cases it solves correctly. For each test case, 50% marks are assigned for correctly deciding whether all jobs can be scheduled (0/1 in line 1 of output) and rest 50% for the optimal cost.