Weighted Interval Scheduling

Project 5 120 points

Due Thursday November 9

Note. The Weighted Interval Scheduling problem is described in the posted file
    Weighted Interval Scheduling.ppt
Memoization is described in the above file and more completely in the posted file
    Chap04 Memoization.pdf
Both of these files are posted in the folder "Lecture Notes".

This project must be done alone.

The **Weighted Interval Scheduling** problem is this: Given a set of weighted intervals, choose a set of non-overlapping intervals such that the total weight is maximal. You may think of the "weight" as the profit for doing the work in the given interval

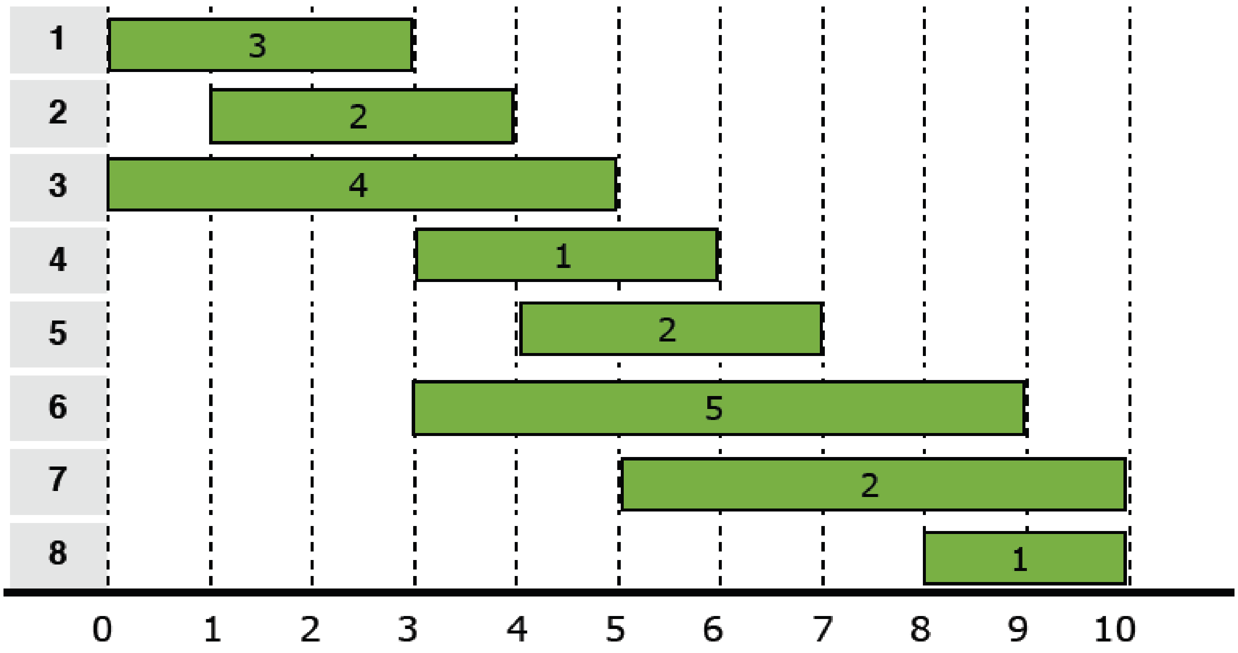
A weighted interval x can be represented by a triple

$x = (s, f, v)$,

where

s = start time of x,   f = finish time of x,   v = weight or profit of this job

For example, consider the test case for Weighted Interval Scheduling problem depicted below:



These weighted intervals can be represented by the triples

(0,3,3) (1,4,2) (0,5,4) (3,6,1) (4,7,2) (3,9,5) (5,10,2) (8,10,1)

Write a program to compute a solution to the Weighted Interval Scheduling problem.

Your program must read in a set of weighted intervals. Each interval should be entered as 3 integers. The intervals must be given in a textfile and also be entered in increasing order of finish time. In the above case we would have

```
0 3 3   1 4 2   0 5 4   3 6 1     . . .
```

The program must print out the value of the total weight or profit of the optimum solution and the indices of the jobs. The output must be in the following format (the output below does NOT have the correct result).

```
Optimum profit: 7
Usng Jobs: 2 5
```

The program MUST use recursion. An iterative solution will not receive full credit. Use of memoization will receive 20 points extra credit.

You must submit a run using the example above. You must also submit a run using the following sample data set:

Number of Jobs n = 4

Job Details {Start Time, Finish Time, Profit}

Job 1: {1, 2, 50}
Job 2: {3, 5, 20}
Job 3: {6, 19, 100}
Job 4: {2, 100, 200}

In both cases you must submit a printout of the input file used.