HW-6 CSL2020 (Jan 2021)

Assigned: 27 Jan 2021 Submission deadline: 3 Feb 2021 (Wednesday) 9 AM Marks: 8

Submit two C source files with names: hw6_q1_roll_number.c and hw6_q2_roll_number.c, and an additional pdf file as mentioned in question 1 below.

1. There is a single classroom but there are N professors interested in using it. The professors have id numbers 1,2,3…,N and we denote any data for their class by using their id number as a subscript (such as x_i). Their classes run for time t_i. There is no fixed start time for their classes, but the professors have other meetings (in fact, I bet, they have plenty of them), and they would like to finish their classes by a preferred deadline of d_i. Given a specific input of (t_i, d_i) pairs, it may not be feasible to provide a schedule which is permissible under the given conditions. For example, (t_1=2, d_1=9), (t_2=5, d_2=6), (t_3=3, d_3=3). The third professor would like to finish her class of 3 time-units by time=3. But if that is permitted then the second professor can't finish his class of 5 time-units by time=6. (Time starts from 0. We assume that the next class can start immediately after the previous one is finished. No two classes can overlap.)

Hence, we relax the conditions a little, and allow delays beyond their preferred finish time. But we don't want to annoy professors (who would?). Hence, we permit delays beyond their preferred times but with a method to minimize the displeasure of the professors. Let us denote the finish time of the class by professor *i* as f_i. We measure the annoyance of professor *i* by the following function:

annoyance$_i$ = max(f_i - d_i , 0) (i.e. no annoyance if the class finishes by the preferred time).

Your goals is to come up with an algorithm which minimizes the ~~sum~~ maximum of annoyance ~~of~~ among all the professors. You are required to schedule all the classes (i.e. none of them can be skipped).

Input: You will read the input in the following way: (spaces between t's and d's, and new line for each class).
N
t_1 d_1
t_2 d_2
…
t_N d_N

Output: Print the order of classes taken by the professors in one line, and the total annoyance you computed in the next line. (e.g. in the given example, the output will be: first line: 3, 2, 1 and the second line: 3. This is because the 2nd class will start at 3 and finish at 8 thus having an annoyance of 2. And the 1st class will start at 8 and finish at 10 having an annoyance of 1. This scheduling produces a total annoyance of 3.) [3 Marks]

You also need to provide a pdf file explaining the algorithm used, the cost of the algorithm and why is this algorithm optimal. Name the file as roll_number.pdf (Write the analysis and the proof of optimality of your algorithm formally). [2 Marks]

2. Two players Akshara and Bharat play a game in which they take turns to make their moves. The game starts with Akshara deciding to collect some numbers from a given sequence. The rules of the game state that the player making the move can collect only one number from the given sequence, and that the selected number must be either at the beginning or at the end of the sequence. Once the player collects the number, the opponent gets to make a move. Every time a player collects the number, this specific number is removed from the sequence. The game stops when the sequence is empty. The winner of the game is the player who has collected a larger sum.

For example, if the given sequence is 3,5,2,1 and Akshara makes the first move. Then she can pick either 3 or 1. Let us suppose she picks 1. Then Bharat gets to move. The remaining sequence for him is 3,5,2. He may decide to pick 3. Then Akshara picks 5 from the sequence 5,2. And finally, Bharat picks the only remaining number 2. The game ends here. Akshara is the winner with her numbers summing to 6, while Bharat could only reach to 5.

Note that if Akshara had picked 3 in her first move then Bharat would have picked 5. And this would have allowed Bharat to win the game. Therefore, Akshara made a good decision by not going for the first number in the sequence on her first move. Hence, by using an “optimal strategy”, Akshara collected 6 points.

Your aim is to develop an efficient algorithm for maximizing the sum of numbers collected by Akshara on a given sequence of numbers, assuming that she makes the first move. You can assume that both the players are playing “optimally”, i.e. trying their best to get the larger sum for themselves.

<u>Input</u>:

A sequence of numbers given in one line. (Some of you were having difficulties with reading inputs in the previous homework. In order to not bother with reading commas, we assume that the input in the following format:

```
N  x1  x2  …  xN
```

Here, N is the number of elements and this is followed by the actual numbers. Further, all the numbers are positive integers. However, they need not be distinct.

<u>Output</u>:

Total sum collected by Akshara. And the sequence of her moves. For instance, the given example will result in an output of:

```
Sum = 6

Steps = 1, 5
```

[3 Marks]