
CS29003 ALGORITHMS LABORATORY
ASSIGNMENT 6 (Greedy Algorithm)
Date: 15th October, 2020

Important Instructions

1. **List of Input Files:** input.txt
2. **Output Files to be submitted:** ROLLNO_GroupNO_A6.c/.cpp
3. Files must be read using file handling APIs of C/C++. Reading through input redirection isn't allowed.
4. You are to **stick to the file input output formats strictly** as per the instructions.
5. Submission through **.zip files are not allowed**.
6. Write your name and roll number at the beginning of your program.
7. Do not use any global variable unless you are explicitly instructed so.
8. Use proper indentation in your code.
9. Please follow all the guidelines. Failing to do so will cause you to lose marks.
10. **There will be part marking.**

Problem Statement

In your city, there is a grocery store that is quite popular. People can order the items online on a given day and collect the parcel next day. On a given day, assume that $2 \leq n \leq 8$ people have given their orders. With the order, they also mention the time window on the next day, when they can collect the parcel. This time window is specified by two integers a_i, b_i , which give the beginning and end of a closed interval $[a_i, b_i]$ during which the person can visit the shop. The numbers a_i and b_i are specified in minutes and satisfy $0 \leq a_i \leq b_i \leq 1440$. Assume that the time taken to collect the parcel is negligible. Further, a constraint due to Covid-19 scenario is that the visiting times should be stretched out as much as possible so that the minimum achievable time gap between successive visits is as large as possible. For example, if three people visit the shop at 10:00 am, 10:05 am, and 10:15 am, then the smallest gap is five minutes, which occurs between the first two people. Not all gaps have to be the same, but the smallest gap should be as large as possible!

See Figure 1 for an illustration. Each line denotes the interval specified by a person. In the leftmost example, there are three people with time windows of $[0, 10]$, $[5, 15]$ and $[10, 15]$, respectively. To maximize the minimum achievable gap, their visits need to be scheduled at times $[0 : 00, 7 : 30, 15 : 00]$ as denoted by the stars, which gives a minimum gap of 7 : 30 minutes (**read 7 minutes 30 seconds**). You can verify that you cannot get a larger gap.

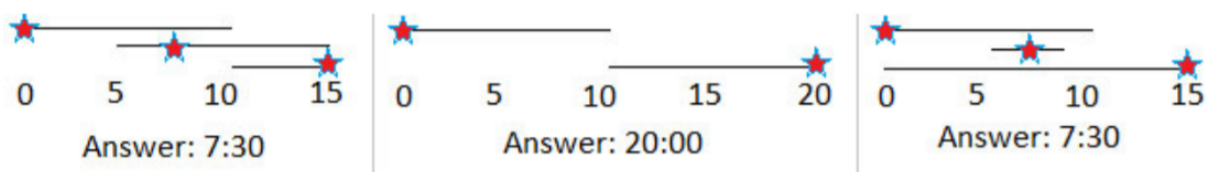


Figure 1: Example Scenarios (answers are given in **mins:secs** format).

Write a `schedule_visits()` function to implement the greedy algorithm. Note that the inputs are not sorted in anyway and if you need to sort you should only use an $O(n \log n)$ algorithm.

1. Compute an order for the visits that respects these time windows given the constraint of maximising the minimum achievable time gap.
 2. Print the answer split into minutes and seconds, rounded to the closest second.
- Hint 1: Try to draw parallels with the activity scheduling problem.
- Hint 2: For each specific visit order, we want to know the largest possible visit window. Suppose we use a certain window length L . We can greedily check whether this L is feasible by forcing the first person to visit as soon as possible and the subsequent persons to visit in `max(a[that person], previous person's arrival time + L)`.

Read the input file as follows

1. The first line contains the number of persons n (assume $2 \leq n \leq 8$)
2. Read a_i, b_i respectively for each person.
 n
 a_1
 b_1
 a_2
 b_2
...so on

Sample 1: input file

FILE: *input1.txt* _____

4
0
8
2
4
3
9
10
20

Sample 2: input file

FILE: *input2.txt* _____

3
0
12
0
2
13
50

You have to print the followings exactly in the same way as shown in the sample output file

1. The first line contains the time gap split into minutes and seconds.
2. The second line prints the order for the visit of the persons.

Sample 1: output file

FILE: *output1.txt* _____

4:00
0 1 2 3

Sample 2: output file

FILE: *output2.txt* _____

12:00
1 0 2

Notes

1. Make sure to read inputs from a file named *“input1.txt”*. At the time of evaluation, the input data might be different from the one given here.
2. You need not to submit *“output1.txt”*, but your code should write the output in the given format in a file named *“output1.txt”*.