CS 325

Homework 4

Problem 1 (5 points)

Road Trip: Suppose you are going on a road trip with friends. Unfortunately, your headlights are broken, so you can only drive in the daytime. Therefore, on any given day you can drive no more than d miles. You have a map with n different hotels and the distances from your start point to each hotel $x_1 < x_2 < ... < x_n$. Your final destination is the last hotel. Describe an efficient greedy algorithm that determines which hotels you should stay in if you want to minimize the number of days it takes you to get to your destination. What is the running time of your algorithm?

Problem 2 (5 points)

Scheduling jobs with penalties: For each $1 \leq i \leq n$ job j_i is given by two numbers d_i and p_i, where d_i is the deadline and p_i is the penalty. The length of each job is equal to 1 minute and once the job starts it cannot be stopped until completed. We want to schedule all jobs, but only one job can run at any given time. If job i does not complete on or before its deadline, we will pay its penalty p_i. Design a greedy algorithm to find a schedule such that all jobs are completed and the sum of all penalties is minimized. What is the running time of your algorithm?

Problem 3 (5 points)

Activity Selection Last-to-Start: Suppose that instead of always selecting the first activity to finish, we instead select the last activity to start that is compatible with all previously selected activities. Describe how this approach is a greedy algorithm, and prove that it yields an optimal solution.

Problem 4 (15 points)

Consider the activity selection last-to-start algorithm described in problem 3.

(a) Write an efficient algorithm (verbal description and pseudocode).

(b) What is theoretical running time of your algorithm?

(c) Implement your algorithm by writing a program named "activity". The program should read input from a file named "act.txt". The first line of the file contains the number of sets. For each set, the number of activities in the set is listed, followed by lines containing the activity number, start time and finish time.

If you use a sorting function in your program you will need to write that yourself. You can use any of the sorting algorithms from previous homework assignments or implement another sorting algorithm. Include the time to sort in your overall running time.

Sample Input: **(*comments* are not part of the file, they are added for clarification)**

```
2            // 2 sets
11           // 11 activities in set 1
1 1 4        // activity number 1, start time 1, finish time 4
2 3 5        // activity number 2, start time 3, finish time 5
3 0 6        // ...
4 5 7
5 3 9
6 5 9
7 6 10
8 8 11
9 8 12
10 2 14
11 12 16
3            // 3 activities in set 2
3 6 8        // activity number 3, start time 6, finish time 8
1 7 9        // ...
2 1 2
```

In the above example the first activity set contains 11 activities with activity 1 starting at time 1 and finishing at time 4, activity 2 starting at time 3 and finishing at time 5, etc. The second activity set contains 3 activities with activity 3 starting at time 6 and finishing at time 8 etc. The activities in the file are not in any sorted order.
Your results including the number of activities selected and their order **should be outputted to the terminal**.

Sample Output for the above example:
Set 1
Number of activities selected = 4
Activities: 2 4 9 11
Set 2
Number of activities selected = 2
Activities: 2 1

Note: There is an alternative optimal solution for Set 1. Since activities 8 and 9 have the same start time of 8, a2 a4 a8 a11 would be an alternative solution. Your program only needs to find one of the optimal solutions. For either solution the activities differ from the solution presented in the text which uses the earliest-finish time criteria.