

CS 325- Homework Assignment 3

Problem 1: 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 $x_1 < x_2 < \dots < x_n$. Your final destination is the last hotel.

- Design an efficient greedy algorithm to determine which hotels you should stay in if you want to minimize the number of days it takes you to get to your destination. Provide a description and pseudocode.
- What is the theoretical running time of your algorithm?

Problem 2: CLRS 16-1-2 Activity Selection Last-to-Start Greedy Criteria

To solve the activity-selection problem (16.1 CLRS), suppose that instead of always selecting the first activity to finish, we 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 3: Activity Selection Last-to-Start Implementation

You may use any language you choose to implement the activity selection last-to-start algorithm described in problem 2. Name your program `activity.**` where `**` = `ccp`, `c` or `py`. 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. If you sort the activities include the time to sort in your theoretical running time.

The program should read input from a file named `act.txt`. The file contains lists of activity sets with number of activities in the set in the first line followed by lines containing the activity number, start time & finish time.

Example `act.txt`:

```
11
1 1 4
2 3 5
3 0 6
4 5 7
5 3 9
6 4 9
7 6 10
8 9 11
9 8 12
10 2 14
11 12 16
3
3 6 8
```

CS 325- Homework Assignment 3

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. For the above example the results are:

Set 1
Maximum number of activities = 4
2 4 8 11

Set 2
Maximum number of activities = 2
2 1

In the write up submitted in Canvas include a description of your algorithm, pseudocode and analysis of the theoretical running time. *You do not need to collected experimental running times.*

In TEACH submit a .zip file containing:

activity.**
act.txt
solution.txt
HW3_**.sh (script file)