CSCI 3104 PS3a

Jonathan Phouminh

TOTAL POINTS

8.5 / 9

QUESTION 1

3 pts

1.1 1 / 1

✓ **+ 0.5 pts** **Size of the array**

✓ **+ 0.5 pts** **list of events**

+ 0 pts incorrect or not attempted

+ 0 pts The question asked to find the maximum number of events that need to selected

1.2 2 / 2

✓ **+ 2 pts** **Correct usage of the Greedy Algorithm and compared to the optimal solution. Good work !!!**

+ 1.5 pts Correct usage of Greedy Algorithm but the comparison of size to optimal solution size of part-a is missing.

+ 0.5 pts Compared with the Greedy Algorithm

+ 1 pts Minor mistakes applying Greedy Algorithm

+ 0 pts Empty or incomplete solution submitted

QUESTION 2

3 pts

2.1 1 / 1

✓ **+ 0.5 pts** **Correct maximum size**

✓ **+ 0.5 pts** **Correct list of events**

+ 0 pts Incorrect answer

+ 0.5 pts Partially correct

2.2 2 / 2

✓ **+ 0.5 pts** **Correct comparison of greedy solution with optimal answer**

✓ **+ 1.5 pts** **Backup solution with correct intervals found with greedy algorithm and correct reasoning**

+ 1 pts Partially correct application of greedy algorithm

+ 0 pts Incorrect

QUESTION 3

3 pts

3.1 1 / 1

✓ **- 0 pts** **Correct**

- 0.5 pts Got the event set wrong

- 0.5 pts Got the size of the set wrong

- 0 pts No need to explain this thoroughly. All I needed was the size and events

- 0.2 pts Give a construction

3.2 1.5 / 2

+ 1.5 pts Using the greedy algorithm correctly

✓ **+ 1 pts** **Minor mistakes applying the greedy algorithm**

✓ **+ 0.5 pts** **Get the not optimal conclusion correctly.**

+ 0 pts Incorrect/Not attempted.

You should choose 8 first.

Name: Jonathan Phouminh

ID: 106054641

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**
Problem Set 3a (9 points) **Fall 2019, CU-Boulder**

Instructions for submitting your solution:

- The solutions **should be typed** and we cannot accept hand-written solutions. Here's a short intro to Latex.
- You should submit your work through **Gradescope** only.
- If you don't have an account on it, sign up for one using your CU email. You should have gotten an email to sign up. If your name based CU email doesn't work, try the identikey@colorado.edu version.
- Gradescope will only accept **.pdf** files (except for code files that should be submitted separately on Gradescope if a problem set has them) and **try to fit your work in the box provided**.
- You cannot submit a pdf which has less pages than what we provided you as Gradescope won't allow it.
- Verbal reasoning is typically insufficient for full credit. Instead, write a logical argument, in the style of a mathematical proof.
- For every problem in this class, you must justify your answer: show how you arrived at it and why it is correct. If there are assumptions you need to make along the way, state those clearly.
- You may work with other students. However, **all solutions must be written independently and in your own words.** Referencing solutions of any sort is strictly prohibited. You must explicitly cite any sources, as well as any collaborators.

Name: Jonathan Phouminh

ID: 106054641

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**

Problem Set 3a (9 points) **Fall 2019, CU-Boulder**

1. Suppose we have a number of events m_i. Each event starts at time s_i and finishes at time e_i, where $0 \leq s_i < e_i$. We represent the event m_i with the closed interval $[s_i, e_i]$. Our goal is to construct a maximum size set of events, where no two events in the set overlap.

   Suppose the following intervals are provided.

Event Index	Interval
1	$[1, 2]$
2	$[3, 4]$
3	$[5, 6]$
4	$[7, 8]$
5	$[0, 20]$.

   (a) (1 pt) What is the maximum size set of events that can be selected such that no two events in the set overlap? Include the list of the events selected in your answer.

   Solution.

Event Index	Interval
1	$[1, 2]$
2	$[3, 4]$
3	$[5, 6]$
4	$[7, 8]$

   Maximum size of set = 4

(b) (2 pt) Suppose we sort the intervals in ascending order by start time. Consider a greedy algorithm that selects the next event based on earliest start time, so long as the interval selected does not conflict with any previously selected interval. Using the intervals provided, show that this greedy algorithm fails to provide a maximum size set of events, where no two events in the set overlap. That is, the solution returned by this greedy algorithm is not optimal.

Solution.

Optimal solutions in this problem is to provide the maximum number of events in the solution set, if we made our greedy algorithm to compare events based on starting time we will see that the algorithm fails to provide the optimal solution because it would first choose event 5 which would run until finish time 20 but we will have missed out on events 1 2 3 and 4. Therefore this algorithm is not optimal.

$$|solutionset| \neq |optimalset|$$

Name: Jonathan Phouminh

ID: 106054641

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**

Problem Set 3a (9 points) **Fall 2019, CU-Boulder**

2. Using the same definition in Problem 1, suppose the following intervals are provided.

Event Index	Interval
1	$[1, 10]$
2	$[11, 20]$
3	$[21, 30]$
4	$[9, 12]$
5	$[19, 22]$.

(a) (1 pt) What is the maximum size set of events that can be selected such that no two events in the set overlap? Include the list of the events selected in your answer.

Solution.

Event Index	Interval
1	$[1, 10]$
2	$[11, 20]$
3	$[21, 30]$

Max number of events = 3

(b) (2 pt) Suppose we sort the intervals in ascending order by interval length. For events with the same length, order by start time. Consider a greedy algorithm that selects the next interval based on the smallest interval length, so long as the interval selected does not conflict with any previously selected interval. Using the intervals provided, show that this greedy algorithm fails to provide a maximum size set of events, where no two events in the set overlap. That is, the solution returned by this greedy algorithm is not optimal.

Solution.

Event Index	Interval
4	$[9, 12]$
5	$[19, 22]$

Based on this greedy algorithm we can see that we fall short of the optimal solution because although picking the smallest interval events would seem to work, we actually end up falling well short of the optimal solution set size. More specifically we fall 2 events short of the optimal solution set size. Thus

$$|solutionset| \neq |optimalset|$$

Name: Jonathan Phouminh

ID: 106054641

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**

Problem Set 3a (9 points) **Fall 2019, CU-Boulder**

3. Consider again the same scenario as in Problems 1 and 2, and suppose the following intervals are provided.

Event Index	Interval
1	$[1, 3]$
2	$[4, 6]$
3	$[7, 9]$
4	$[10, 12]$
5	$[2, 5]$
6	$[2, 5]$
7	$[2, 5]$
8	$[5.5, 7.5]$
9	$[8, 11]$
10	$[8, 11]$
11	$[8, 11]$

(a) (1 pt) What is the maximum size set of events that can be selected such that no two events in the set overlap? Include the list of the events selected in your answer.

Solution.

Event Index	Interval
1	$[1, 3]$
2	$[4, 6]$
3	$[7, 9]$
4	$[10, 12]$

Maximum solution set size = 4

(b) (2 pts) Let c_i denote the number of intervals on our list in which interval i conflicts. For example, interval 1 participates in 3 conflicts: with intervals $5, 6$, and 7. So $c_1 = 3$.

Suppose we sort the intervals in ascending order based on the number of conflicts. So if $c_i < c_j$, then interval i comes before interval j. Consider a greedy algorithm that selects the next interval based on the smallest number of conflicts, so long as the interval selected does not conflict with any previously selected interval. Using the intervals provided, show that this greedy algorithm fails to provide a maximum size set of events, where no two events in the set overlap. That is, the solution returned by this greedy algorithm is not optimal.

Solution.

Event Index	Interval	Conflict Count
1	$[1, 3]$	3
2	$[4, 6]$	4
3	$[7, 9]$	3
4	$[10, 12]$	3
5	$[2, 5]$	4
6	$[2, 5]$	4
7	$[2, 5]$	4
8	$[5.5, 7.5]$	2
9	$[8, 11]$	4
10	$[8, 11]$	4
11	$[8, 11]$	4

$Solutionset = [1, 3, 4]$ We can see that there are only three events in the solution set based on this greedy method and thus $|solutionset| \neq |optimalset|$ therefore this algorithm is not optimal.