Name: Nikolaas Bender
ID: 106977096

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: Nikolaas Bender
ID: 106977096

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. 4 events is the maximum number of events that can be selected without any events overlapping
   the optimal set is:
   $[1, 2]$
   $[3, 4]$
   $[5, 6]$
   $[7, 8]$
   This is the optimal set because the set has 4 events while the alternative only contains 1 event as that one event is [0, 20] which if selected no other events from the original set would be allowed.
   this seems too easy

(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. The proposed greedy algorithm would select the event index 5 with interval $[0, 20]$
this is not the optimal solution because by choosing a different selection policy we can schedule 4 events, event indices 1, 2, 3, 4. This more optimal policy was found through experimentation though.

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. the maximum we can select is 3 intervals
$[1, 10]$
$[11, 20]$
$[21, 30]$
as they do not over lap

(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. to begin thinking of the solution:

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

This algorithm would choose
$[9, 12]$
$[19, 22]$
which is not optimal because we can choose a policy that will allow for 3 sets instead of 2

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. the optimal solution is
$[1, 3]$
$[4, 6]$
$[7, 9]$
$[10, 12]$
which is only marginally better than
$[2, 5]$
$[5.5, 7.5]$
$[8, 11]$

(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. to start thinking of the solution

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

this greedy algorithm would select
$[1, 3]$
$[5.5, 7.7]$
$[8, 11]$
When the interval of $[5.5, 7.5]$ is selected then all intervals before it should be thrown out, however Rhonda told me otherwise.
Then $[8, 11]$ is selected as it is one of two remaining compatible slots but it starts earlier than $[10, 12]$