CSCI 3104 Spring 2023
Instructors: Ryan Layer and Chandra Kanth Nagesh

Homework 4

Due Date .... February 9, 2023
Name .... **Blake Raphael**
Student ID **109752312**
Collaborators .... **Alex Barry and Brody Cyphers**

Contents

1 Instructions

- The solutions **should be typed**, using proper mathematical notation. We cannot accept hand-written solutions. Here's a short intro to LaTeX.

- You should submit your work through the **class Gradescope page** only (linked from Canvas). Please submit one PDF file, compiled using this LaTeX template.

- You may not need a full page for your solutions; pagebreaks are there to help Gradescope automatically find where each problem is. Even if you do not attempt every problem, please submit this document with no fewer pages than the blank template (or Gradescope has issues with it).

- You are welcome and encouraged to collaborate with your classmates, as well as consult outside resources. You must **cite your sources in this document. Copying from any source is an Honor Code violation. Furthermore, all submissions must be in your own words and reflect your understanding of the material.** If there is any confusion about this policy, it is your responsibility to clarify before the due date.

- Posting to **any** service including, but not limited to Chegg, Reddit, StackExchange, etc., for help on an assignment is a violation of the Honor Code.

- You **must** virtually sign the Honor Code (see Section 2). Failure to do so will result in your assignment not being graded.

2 Honor Code (Make Sure to Virtually Sign)

- My submission is in my own words and reflects my understanding of the material.
- Any collaborations and external sources have been clearly cited in this document.
- I have not posted to external services including, but not limited to Chegg, Reddit, StackExchange, etc.
- I have neither copied nor provided others solutions they can copy.

Agreed (I agree to the following, Blake Raphael). □

3 Standard 4 – Examples Where Greedy Algorithms Fail

3.3 Problem 1 (2 points)

Problem 1. Recall the Interval Scheduling problem, where we take as input a set of intervals $\mathcal{I}$. The goal is to find a maximum-sized set $S \subseteq \mathcal{I}$, where no two intervals in S intersect. Consider the greedy algorithm where we place all of the intervals of $\mathcal{I}$ into a priority queue, ordered earliest start time to latest start time. We then construct a set S by adding intervals to S as we poll them from the priority queue, provided the element we polled does not intersect with any interval already in S.

Provide an example with at least 5 intervals where this algorithm fails to yield a maximum-sized set of pairwise non-overlapping intervals. Clearly specify both the set S that the algorithm constructs, as well a larger set of pairwise non-overlapping intervals.

You may explicitly specify the intervals by their start and end times (such as in the examples from class) or by drawing them. **If you draw them, please make it very clear whether two intervals overlap.** You are welcome to hand-draw and embed an image, provided it is legible and we do not have to rotate our screens to grade your work. Your justification should still be typed. If you would prefer to draw the intervals using LaTeX, we have provided sample code below.

Answer. Consider the set X composed of 8 start and end time intervals $[start, end]$.
Set X is composed of the following intervals:

Interval	[Start, End]
a	[8, 12:30]
b	[9, 10:30]
c	[10, 11]
d	[10, 1:30]
e	[11, 1]
f	[11:30, 2]
g	[12, 3]
h	[1:30, 4]

If we use a greedy algorithm taking the earliest, non-overlapping start time first, we get set $S = a[8, 12:30], h[1:30, 4]$.

This is **NOT** the max-size, non-overlapping set. That would be set M where $M = b[9, 10:30], e[11, 1], h[1:30, 4]$. This shows an example where a greedy algorithm fails.

□

3.4 Problem 2 (2 points)

Problem 2. Consider now the Weighted Interval Scheduling problem, where each interval i is specified by

$$([\text{start}_i, \text{end}_i], \text{weight}_i).$$

Here, the weight is an assigned value that is independent of the length $\text{end}_i - \text{start}_i$. Here, you may assume $\text{weight}_i > 0$. We seek a set S of pairwise non-overlapping intervals that maximizes $\sum_{i \in S} \text{weight}_i$. That is, rather than maximizing the number of intervals, we are seeking to maximize the sum of the weights.

Consider a greedy algorithm which works identically as in Problem 1. Draw an example with at least 5 appointments where this algorithm fails. Show the order in which the algorithm selects the intervals, and also show a subset with larger weight of non-overlapping intervals than the subset output by the greedy algorithm. The same comments apply here as for Problem 1 in terms of level of explanation.

Answer. Consider the set X composed of 8 start and end time intervals $([start, end], weight)$.
Set X is composed of the following intervals:

Interval	**[Start, End]**
a	([8, 12:30], 1)
b	([9, 10:30], 11)
c	([10, 11], 2)
d	([10, 1:30], 2)
e	([11, 1], 4)
f	([11:30, 2], 2)
g	([12, 3], 1)
h	([1:30, 4], 7)

Suppose we use a greedy algorithm taking the earliest, non-overlapping start time first.

This algorithm would add interval a to set S first since it starts earliest.

The algorithm would then check b and find that it overlaps with a, so then it would move on.

The algorithm will check every interval c-g and find that all of these overlap with a.

Finally, it will check interval h and find that it does not overlap, and it is the earliest available start time since it is the last interval, so it can be added to set S.

We get set $S = a([8, 12:30], 1), h([1:30, 4], 7)$.

This gives us a total weight of 8.

Set S is **NOT** the highest-weight, non-overlapping set.

That would be set M where $M = b([9, 10:30], 11), e([11, 1], 4), h([1:30, 4], 7)$.

This gives us a total weight of 22.

This shows an example where a greedy algorithm again fails.

□