

Problem Set 4

Brian E. Peterson

06/16/2021

Due Date 06/16/2021
Name **Brian E. Peterson**
Student ID **109067890**
Collaborators **N/A**

Contents

1	Instructions	1
2	Honor Code (Make Sure to Virtually Sign the Honor Pledge)	2
3	Standard 4- Examples Where Greedy Algorithms Fail	3
3.1	Problem 2	3
3.2	Problem 3	4
3.3	Problem 4	5
4	Standard 5- Exchange Arguments	6
4.1	Problem 5	6
4.2	Problem 6	7
4.2.1	Problem 6(a)	7
4.2.2	Problem 6(b)	8

1 Instructions

- The solutions **must 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 Canvas page** only. 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 the Honor Pledge)

Problem 1. On my honor, my submission reflects the following:

- 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.

In the specified region below, clearly indicate that you have upheld the Honor Code. Then type your name.

Honor Pledge. • 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.

Brian E. Peterson

□

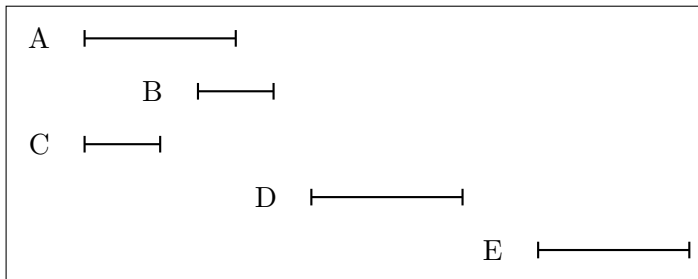
3 Standard 4- Examples Where Greedy Algorithms Fail

3.1 Problem 2

Problem 2. 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 Example 35 of the course notes) or by drawing them. 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 L^AT_EX, we have provided sample code below.



Answer.

Given this set of intervals, \mathcal{I} , the above described algorithm would construct the following set, S :

$$[A, D, E]$$

The algorithm would construct this set because A and C have the same start time, but A appears first. If we assume the priority queue would contain first A, then C, followed by B, D, and E, then the initial choice of A would result in not including 2 intervals which could be exchanged for A, thus finding a non-maximum-sized set. However, the maximum-sized set of pairwise non-overlapping intervals would actually be:

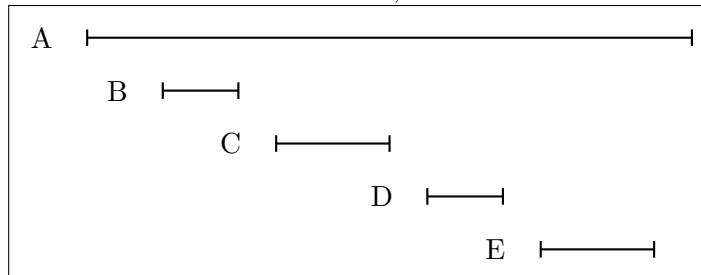
$$[C, B, D, E]$$

□

3.2 Problem 3

Problem 3. Recall that the length of an interval $[s, f]$ is $f - s$. Consider a greedy algorithm which works identically as in Problem 2, except we sort the intervals from longest to shortest. 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 larger subset of non-overlapping intervals than the subset output by the greedy algorithm. The same comments apply here as for Problem 2 in terms of level of explanation.

Answer. Take this set of intervals, \mathcal{I} :



The algorithm would first produce this ordering:

$$[A, C, E, B, D]$$

Then it would put A into S and then immediately terminate, because all other intervals intersect with A , and A was selected first, precluding adding any more intervals. Yielding a final set S :

$$[A]$$

This is clearly not the maximum-sized set of intervals in \mathcal{I} , which is actually (leaving the sort order as above):

$$[C, E, B, D]$$

□

3.3 Problem 4

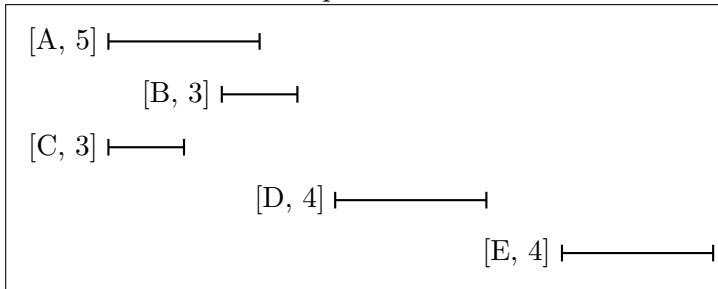
Problem 4. Consider now the Weighted Interval Scheduling problem, where each interval i is specified by

$$([start_i, end_i], weight_i).$$

Here, the weight is an assigned value that is independent of the length $end_i - start_i$. Here, you may assume $weight_i > 0$. We seek a set S of pairwise non-overlapping intervals that maximizes $\sum_{i \in S} weight_i$. 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 2. 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 2 in terms of level of explanation.

Answer. Consider this example:



Let the weight of each interval be the number which follows the interval's label, e.g. $[F, 6]$ has weight 6. This is notably, essentially the same answer I gave for 2. Again, the algorithm sorts these as follows (by start time, not weight – this is the problem specification):

$$[A, C, B, D, E]$$

Upon polling A , then C and B become inadmissible into S because they intersect with A . Adding D and E , this yields:

$$[A, D, E]$$

However, this has weights $5 + 4 + 4 = 13$, whereas the set of intervals with maximum weights is:

$$[C, B, D, E],$$

Having weights $3 + 3 + 4 + 4 = 14$.

□

4 Standard 5- Exchange Arguments

4.1 Problem 5

Problem 5. Recall the Making Change problem, where we have an infinite supply of pennies (worth 1 cent), nickels (worth 5 cents), dimes (worth 10 cents), and quarters (worth 25 cents). We take as input an integer $n \geq 0$. The goal is to make change for n using the fewest number of coins possible.

Prove that in an optimal solution, we use at most 4 pennies.

Proof. Let S be the multiset of coins, with which, we wish to make change for n . Suppose S contains $k > 4$ pennies, so $k \geq 5$. By the division algorithm,

$$k = 5j + r$$

For $k > 4$, we can exchange j , or $\lfloor \frac{k}{5} \rfloor$, quintets of pennies for 1 nickel. This results in $4j$ fewer coins needed to make change for n , as we remove 5 pennies while adding 1 nickel, for each exchange that we make. If $k > 4$, we can make at least one such exchange, since $j > 0$. Therefore, in an optimal change-making solution, we will use at most 4 pennies because we will have made all possible such exchanges which can each remove 4 coins from our total solution, i.e. our solution will have the minimal number of coins (and also the minimum). \square

4.2 Problem 6

Problem 6. Consider the Interval Projection problem, which is defined as follows.

- **Instance:** Let \mathcal{I} be a set of intervals on the real line.
- **Solution:** A minimum sized set S of points on the real line, such that (i) for every interval $[s, f] \in \mathcal{I}$, there exists a point $x \in S$ where x is in the interval $[s, f]$. We call S a *projection set*.

Do the following.

4.2.1 Problem 6(a)

- (a) Find a minimum sized projection set S for the following set of intervals:

$$\mathcal{I} = \{[0, 1], [0.5, 1], [1, 1.5], [1.4, 2], [1.6, 2.3]\}.$$

Answer.

$$S = \{0.5, 1.4, 2.3\}$$

This is a projection set for \mathcal{I} because 0.5 is in both of the first two intervals, 1.4 is contained in the third and fourth, and 2.3 is contained in the last. It's a minimal-sized projection set because any point in $[0.5, 1]$ satisfies the first two intervals (this is their intersection), any point in $[1.4, 1.5]$ satisfies the third and fourth intervals (again, their intersection), and any point in $[1.6, 2.3]$ satisfies the last interval (no change made). However, these intersections of the original intervals are pairwise non-overlapping, and thus no less than 3 points, 1 for each interval, can serve as a projection set for them. This is the size of our given solution, so we do have a minimum-sized projection set.

□

4.2.2 Problem 6(b)

- (b) Fix a set of intervals \mathcal{I} , and let S be a projection set. Prove that there exists a projection set S' where every point $x \in S'$ is the right end-point of interval $[s, f] \in \mathcal{I}$.

Proof. Base case. Consider the case when $\mathcal{I} = \{\}$.

By definition, a projection set S is a set of points on the real line, such that for every interval $[s, f] \in \mathcal{I}$, there exists a point $x \in S$ where x is in the interval $[s, f]$.

In the case of $\mathcal{I} = \{\}$ (the empty set), the only valid projection set is by definition also the empty set. This is indeed a projection set, because for each interval in \mathcal{I} , (i.e. no intervals), there exists a point. Because the premise never holds (there are no intervals), the proposition is always true (there exists a point for each non-existent interval).

For a similar reason, this projection set is also one where every point $x \in S'$ is the right end-point of an interval, i.e. there are no points, and all of them (none) are right end-points of intervals in \mathcal{I} . ✓

Inductive hypothesis. Fix $k > 0$. Assume for $|\mathcal{I}| = k$, there is a projection set S' where every point $x \in S'$ is the right end-point of one of the k intervals in \mathcal{I} .

Inductive step. Consider the $k + 1$ st case.

From the inductive hypothesis (IH), we know there is some set of points, S' , where every point is a right end-point of an interval from \mathcal{I} , and that this constitutes a projection set for \mathcal{I} .

Therefore, we know that all intervals except one new one, just added, are already "represented", and only by their or other right end-points, in some set, S' .

Consider the case where the new interval is entirely contained within another, already "represented" interval. Then we already have a projection set which is valid for the $k + 1$ st case. Without adding any more points, we also know that every point in S' corresponds to the right end-point of some interval. ✓

Consider the case where the interval is not entirely contained within another, already "represented" interval. Then either the right or left or both end-points are outside all of our intervals.

If only the left end-point is outside of our existing intervals, then the right end-point is inside our existing intervals. In this case, we again already have what we need: a representative point for our projection set which is the right end-point of an interval. ✓

If the right end-point is outside our existing intervals, we need to show that there is some new projection set S'' which can contain all the points in S' , plus our new right endpoint.

Suppose the new interval being added to our \mathcal{I} is $[s_i, f_i]$. Then, $S'' = S' \cup f_i$.

Therefore, by induction on $|\mathcal{I}|$, there always exists a projection set S' for \mathcal{I} , where every point $x \in S'$ is the right end-point of some interval $[s, f] \in \mathcal{I}$.

□