CSCI 3104 Spring 2022
Instructors: Profs. Chen and Layer

Problem Set 1

Due Date .... January 25, 2022
Name .... **Julia Troni**
Student ID **109280095**
Collaborators .... **My brain and TAs**

Contents

1 Instructions 1

2 Honor Code (Make Sure to Virtually Sign) 2

3 Standard 1- Proof by Induction 3
3.1 Problem 2 3
3.2 Problem 3 4
3.3 Problem 4 5

4 Standard 2- Examples Where Greedy Algorithms Fail 7
4.1 Problem 5 7
4.2 Problem 6 8

5 Standard 3- Exchange Arguments 9
5.1 Problem 7 9
5.2 Problem 8 10
5.2.1 Problem 8(a) 10
5.2.2 Problem 8(b) 11

6 Standard 4- Huffman coding 12
6.1 Problem 9 12

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 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)

Problem 1.

- 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.

I agree to the above, Julia Troni. $\square$

3 Standard 1- Proof by Induction

3.1 Problem 2

Problem 2. A student is trying to prove by induction that $3^n < n!$ for $n \geq 7$.

Student's Proof. The proof is by induction on $n \geq 7$.

- **Base Case:** When $n = 7$, we have that:

$$\begin{aligned} 3^7 &= 2187 \\ &< 5040 \\ &= 7! \end{aligned}$$

- **Inductive Hypothesis:** Now suppose that for all $k \geq 7$ we have that $3^k < k!$.
- **Inductive Step:** We now consider the $k+1$ case. We have that $3^{k+1} < (k+1)!$. It follows that $3^k < k!$. The result follows by induction.

□

There are two errors in this proof.

(a) The Inductive Hypothesis is not correct. Write an explanation to the student explaining why their Inductive Hypothesis is not correct. [**Note:** You are being asked to explain why the Inductive Hypothesis is wrong, and **not** to rewrite a corrected Inductive Hypothesis.]

Answer. The student incorrectly assumes that $3^k < k!$ holds true for ALL $k \geq 7$. This is incorrect because the student is incorrectly assuming that the entire theorem is true, but they have not yet proven this. At this point the student has only shown that the statement holds for a fixed range ≥ 7. So they should say something like "Fix $k \geq 7$ and suppose that $3^k < k!$. □

(b) The Inductive Step is not correct. Write an explanation to the student explaining why their Inductive Step is not correct. [**Note:** You are being asked to explain why the Inductive Step is wrong, and **not** to rewrite a corrected Inductive Step.]

Answer. The inductive step is incorrect because the student manipulated both sides of the inequality at the same time and did not actually show how $3^{k+1} < (k+1)!$. Instead, they need to start with one side of the inequality and manipulate that one side until they reach the expression on the other side. For example

$$\begin{aligned} 3^{k+1} &< 3^k \cdot 3 \\ &< k! \cdot 3 && \text{(by IH)} \\ &< k! \cdot (k+1) && \text{(since 3 < k+1)} \\ &< (k+1)! \end{aligned}$$

□

3.2 Problem 3

Problem 3. Consider the sequence $T_n,\ n \geq 1$ defined by the following recurrence: $T_1 = T_2 = T_3 = 1$ and $T_n = T_{n-1} + T_{n-2} + T_{n-3}$ for $n \geq 4$.

Prove by induction that $T_n < 2^n$ for all $n \geq 1$.

Proof. Proof by induction on n.

- **Base Cases:**
  $T_1 = 1 < 2^1 = 2$,
  $T_2 = 1 < 2^2 = 4$,
  $T_3 = 1 < 2^3 = 8$

$$\begin{aligned} T_4 &= T_1 + T_2 + T_3 \\ &= 1 + 1 + 1 \\ &= 3 < 2^4 = 16 \end{aligned}$$

  so the base cases hold

- **Inductive Hypothesis:**
  Let $k \geq 1$ and suppose for $1 \leq i \leq k, T_i < 2^i$

- **Inductive Step:** Using that $T_i < 2^i$ for $1 \leq i \leq k$, we will show that $T_{k+1} < 2^{k+1}$

$$\begin{aligned} T_{k+1} &= T_k + T_{k-1} + T_{k-2} \\ &< 2^k + 2^{k-1} + 2^{k-2} && \text{by the Inductive Hypothesis} \\ &< 2^k + 2^k \cdot \frac{1}{2} + 2^k \cdot \frac{1}{4} \\ &< 2^k \cdot (1 + \frac{1}{2} + \frac{1}{4}) \\ &< 2^k \cdot (2) && \text{since } 1.75 < 2 \\ &< 2^{k+1} \end{aligned}$$

  So by strong induction we have that $T_n < 2^n$ for all $n \geq 1$.

□

3.3 Problem 4

Problem 4. The complete, balanced ternary tree of depth d, denoted $\mathcal{T}(d)$, is defined as follows.

- $\mathcal{T}(0)$ consists of a single vertex.
- For $d > 0$, $\mathcal{T}(d)$ is obtained by starting with a single vertex and setting each of its three children to be copies of $\mathcal{T}(d-1)$.

Prove by induction that $\mathcal{T}(d)$ has 3^d leaf nodes. To help clarify the definition of $\mathcal{T}(d)$, illustrations of $\mathcal{T}(0), \mathcal{T}(1)$, and $\mathcal{T}(2)$ are on the next page. [**Note:** $\mathcal{T}(d)$ is a tree and **not** the number of leaves on the tree. Avoid writing $\mathcal{T}(d) = 3^d$, as these data types are incomparable: a tree is not a number.]

Proof. Let $L(d)$ denote the number of leaf nodes on the tree.
I will prove that $\mathcal{T}(d)$ has $L(d) = 3^d$ leaf nodes by induction on the depth of the tree, d.

- **Base Case:**
  $\mathcal{T}(0)$ is a single vertex, so $\mathcal{T}(0)$ has $L(0) = 3^0 = 1$ leaf nodes
  Thus, the base case holds.

- **Inductive Hypothesis:**
  Fix $d \geq 0$ and suppose that $T(d)$ has $L(d) = 3^d$ leaf nodes.

- **Inductive Step:** If $\mathcal{T}(d)$ has $L(d) = 3^d$ leaf nodes, then I will show that $\mathcal{T}(d+1)$ has $L(d+1) = 3^{d+1}$ leaf nodes.

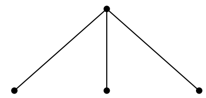
  $\mathcal{T}(d+1)$ has a single root r where each of r's 3 children are copies of $\mathcal{T}(d)$. So, by inductive hypothesis, all 3 copies of $\mathcal{T}(d)$ have $L(d) = 3^d$ leaf nodes. Which is $L(d+1) = 3(3^d)$ in $\mathcal{T}(d+1)$.
  So $\mathcal{T}(d+1)$ has $3(3^d) = 3^{d+1}$ leaf nodes.

  So by induction we have that $\mathcal{T}(d)$ has 3^d leaf nodes.

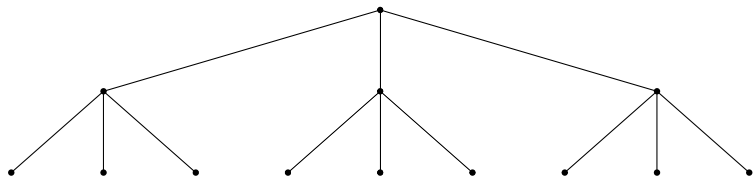
$\square$

Example 1. We have the following:

$\mathcal{T}(0)$.



$\mathcal{T}(1)$.



$\mathcal{T}(2)$.

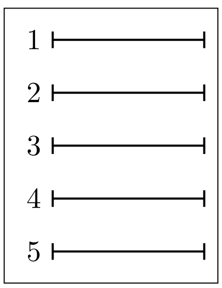
4 Standard 2- Examples Where Greedy Algorithms Fail

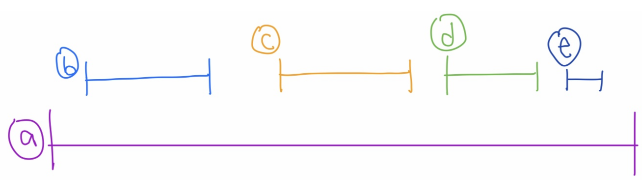
4.1 Problem 5

Problem 5. 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.
For the example given above, the greedy algorithm will fail to yield a maximum sized set of pairwise non-overlapping intervals. Greedy will place the intervals of $\mathcal{I}$ into a priority queue with the order $\{a, b, c, d, e\}$ and produce the set $S = \{a\}$. However, the maximum sized set is clearly $\{b, c, d, e\}$, but greedy algorithm will never yield this result.

□

4.2 Problem 6

Problem 6. 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 5. 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 5 in terms of level of explanation.

Answer.
The greedy algorithm fails for this set of intervals because the greedy algorithm will order the intervals as $\{b, a, c, d, e\}$ then it will produce the set $S_{greedy} = \{b, c, d, e\}$ which $\sum_{i \in S} \text{weight}_i = 14$. However, this does not maximize the weight, and it only maximizes the number of intervals. The correct solution to maximize the weight is the set $S_{maxweight} = \{a\}$ because the$\sum_{i \in S} \text{weight}_i = 50$. and $50 > 14$. □

5 Standard 3- Exchange Arguments

5.1 Problem 7

Problem 7. 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 2 dimes.

Proof. Suppose set I is an optimal solution, meaning it is a set of coins that makes change for n using the fewest number of coins possible.
Now also suppose that I contains > 2 dimes.

The key idea is that we can exchange every trio of 3 dimes for 1 quarter and 1 nickel.
By the division algorithm, $k = 3j + r$ where $j \in \mathbb{N}$ and $r \in \{0, 1, 2\}$
That is, anytime we have $3j$ (a trio of dimes), we can reduce 3 coins into 2 by exchanging the 3 dimes for 1 quarter and 1 nickel.

Seeing as I contains $k > 2$ dimes, we have that $j \geq 1$. So we can exchange $3j$ dimes for $1j$ quarters $+1j$ nickels to make the new solution set I'. Since $2 < 3$, $|I'| = |I| - j < |I|$.

Thus, since we can produce a set I' that uses fewer coins that I, it follows that an optimal solution contains a maximum of 2 dimes.

□

5.2 Problem 8

Problem 8. 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 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*.

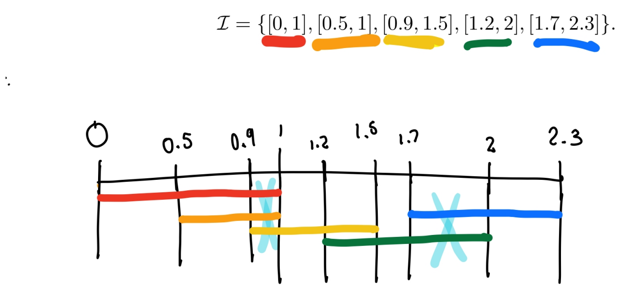
Do the following.

5.2.1 Problem 8(a)

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

$$\mathcal{I} = \{[0, 1], [0.5, 1], [0.9, 1.5], [1.2, 2], [1.7, 2.3]\}.$$

Answer. A projection set S for the intervals is $S = \{0.95, 1.75\}$ which is visualized in the sketch below.



$\square$

5.2.2 Problem 8(b)

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

Proof. Suppose we have some projection set S that projects into the set of intervals I. Let x be a member of S that intersects intervals $[1, 2, ..., k]$ where x is between $[s_1, f_i], [s_2, f_2], ..., [s_k, f_k]$.

Now suppose we exchange x with f_m, which is the endpoint that is the first to finish, and denote this new set S'. S' is still a projection set because f_m still intersects the same number of intervals as x did. Since f_m is the rightmost endpoint of the first interval to finish, by definition it intersects itself. And we know that x intersected between s_m and f_m, so $s_{1,2,...k} \leq x \leq f_m \leq f_{1,2,...,k}$.
So $|S'| = |S|$

Since x is a generic element of S, we can apply this exchange for all elements of S, meaning we can exchange all elements of S with some endpoint, f_y. Thus, we would create S', a projection set where every point $p \in S'$ is the right end-point.

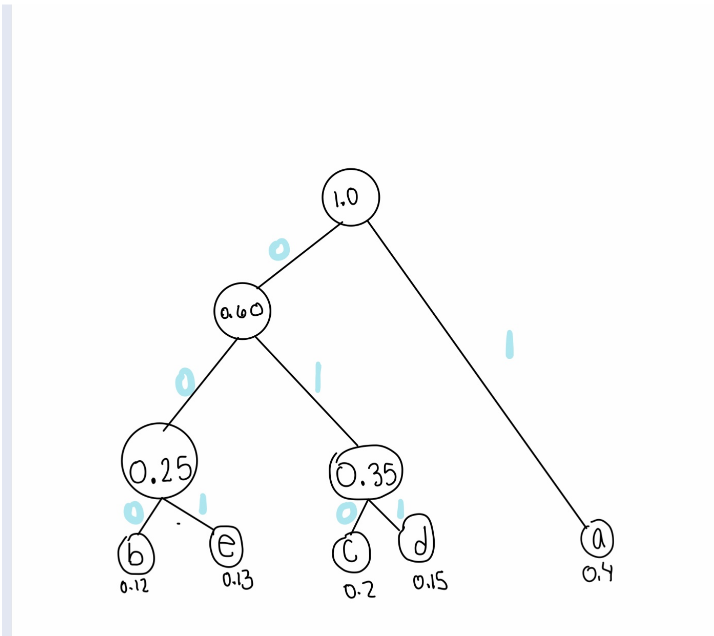
□

6 Standard 4- Huffman coding

6.1 Problem 9

Problem 9. Given an alphabet of five symbols: a, b, c, d and e, with frequencies 0.4, 0.12, 0.2, 0.15, and 0.13 respectively, work out the Huffman codes for the symbols. You need to first show the optimal binary tree you construct, and then write down the corresponding codes.

Answer. The optimal binary tree is as follows



And the corresponding codes are
b: 000
e: 001
c: 010
d: 011
a: 1

□