CSCI 3104 Spring 2022
Instructors: Profs. Chen and Layer

Problem Set 1

Due Date	January 25, 2022
Name	**Alex Ojemann**
Student ID	**109722375**
Collaborators	

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 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.

Agreed (Alex Ojemann). I agree to the above, Alex Ojemann □

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. Their inductive hypothesis assumes that for all $k \geq 7$ that $3^k < k!$. It should instead assume that only for some $k \geq 7$ that $3^k < k!$. Assuming that it's correct for all values of $k \geq 7$ is incorrect because that would be assuming that the proof itself is correct. You can only make the assumption that there exists a k that's greater than or equal to 7 where this is true. □

(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 doesn't mathematically show that $3^k < k!$ follows from $3^{k+1} < (k + 1)!$. He would need to simplify the left side of $3^{k+1} < (k + 1)!$ to $3 * 3^k$ and the right side to $(k + 1) * k!$ and state that since $3 > k + 1$ because we assume $k \geq 7$ in our inductive hypothesis and $3^k < k!$ because we assumed this as well in our inductive hypothesis that $3^{k+1} < (k + 1)!$ must be true because the left side is a product of two numbers that are smaller than the two numbers that the right side is a product of respectively. □

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. The proof is by induction on $n \geq 1$.

- **Base Cases:** When $n = 1$, we have that:

$$\begin{aligned} T_1 &= 1 \\ &< 2 \\ &= 2^1 \end{aligned}$$

  When $n = 2$, we have that:

$$\begin{aligned} T_2 &= 1 \\ &< 4 \\ &= 2^2 \end{aligned}$$

  When $n = 3$, we have that:

$$\begin{aligned} T_3 &= 1 \\ &< 8 \\ &= 2^3 \end{aligned}$$

  When $n = 4$, we have that:

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

- **Inductive Hypothesis:** Now suppose that for some $k \geq 1$ we have that $T_k < 2^k$, $T_{k-1} < 2^{k-1}$ and $T_{k-2} < 2^{k-2}$.

- **Inductive Step:** We now consider the $k + 1$ case. We have that $T_k + 1 < 2^{k+1}$.

$$\begin{aligned} T_k + 1 &= T_k + T_{k-1} + T_{k-2} \\ &< 2^k + 2^{k-1} + 2^{k-2} \\ &< 2^k + 2^{k-1} + 2^{k-2} + 2^{k-2} \\ &= 2^{k+1} \end{aligned}$$

  Here, the second line follows by applying the inductive hypothesis to obtain that $T_k < 2^k$, $T_{k-1} < 2^{k1}$ and $T_{k-2} < 2^{k2}$. The result follows by induction.

□

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.]

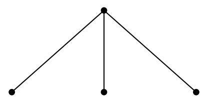
Proof. The proof is by induction on $n \geq 7$.

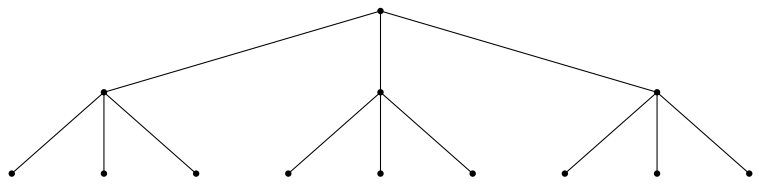
- **Base Case:** When $d = 0$, we have that: $\mathcal{T}(0)$ has $3^0 = 1$ leaf node, as desired.
- **Inductive Hypothesis:** Now suppose that for some $d \geq 0$ we have that $\mathcal{T}(d)$ has 3^d leaf nodes.
- **Inductive Step:** We now consider the $d+1$ case. We have that $\mathcal{T}(d+1)$ should have 3^{d+1} leaf nodes. By construction, $\mathcal{T}(d+1)$ consists of a single root node v, where each of v's three children are copies of $\mathcal{T}(d)$. By the inductive hypothesis, all 3 copies of $\mathcal{T}(d)$ have 3^d leaf nodes. So, $\mathcal{T}(d+1)$ has $3 * 3^d = 3^{d+1}$ leaf nodes as desired. The result follows by induction.

$\square$

Example 1. We have the following:

$\mathcal{T}(0)$.


$\mathcal{T}(1)$.


$\mathcal{T}(2)$.

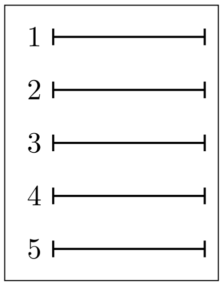
4 Standard 2- Examples Where Greedy Algorithms Fail

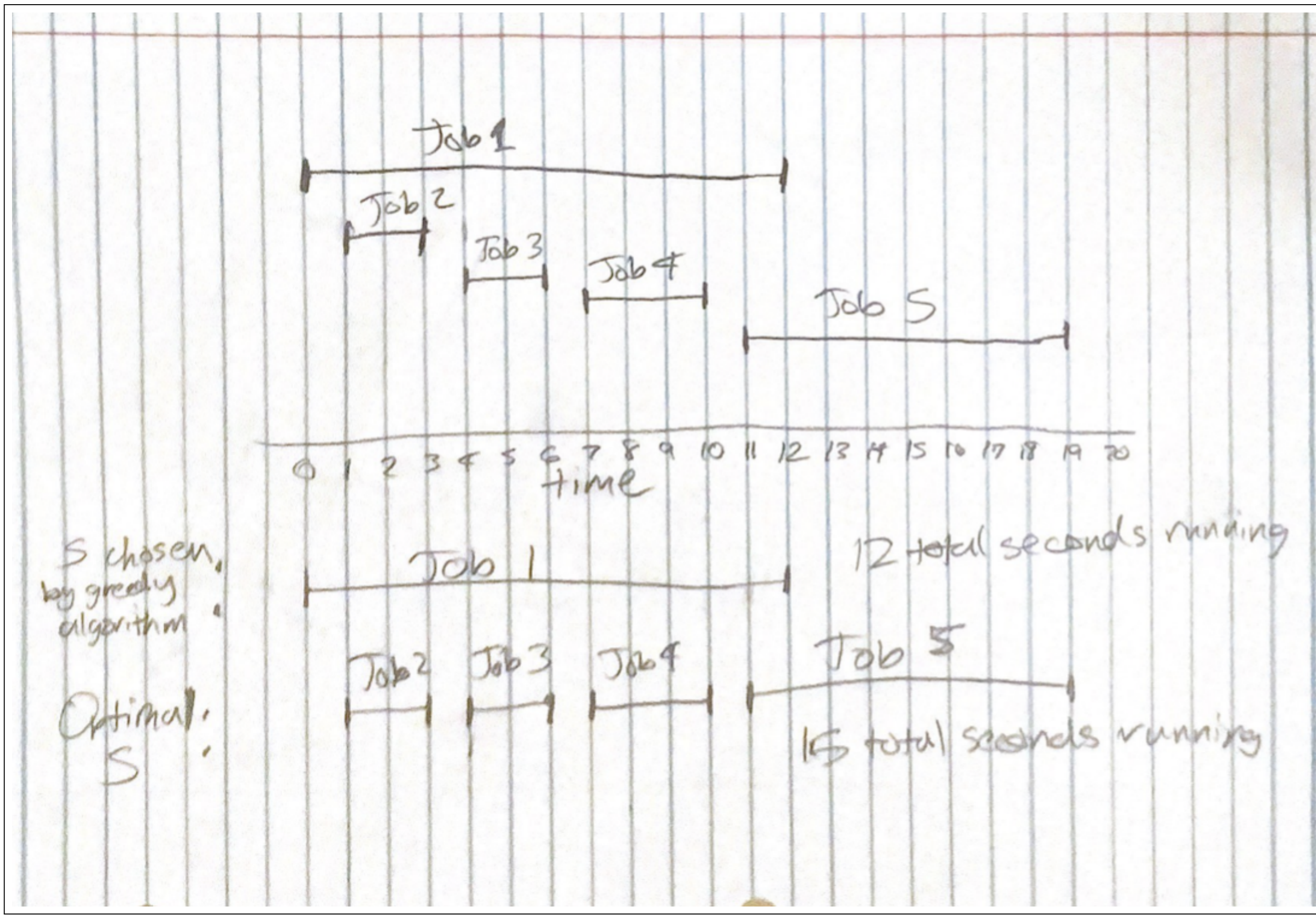
4.1 Problem 5

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. 
Job 1
Job 2
Job 3
Job 4
Job 5
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
time
S chosen by greedy algorithm:
Job 1
12 total seconds running
Optimal: S
Job 2
Job 3
Job 4
Job 5
15 total seconds running

□

4.2 Problem 6

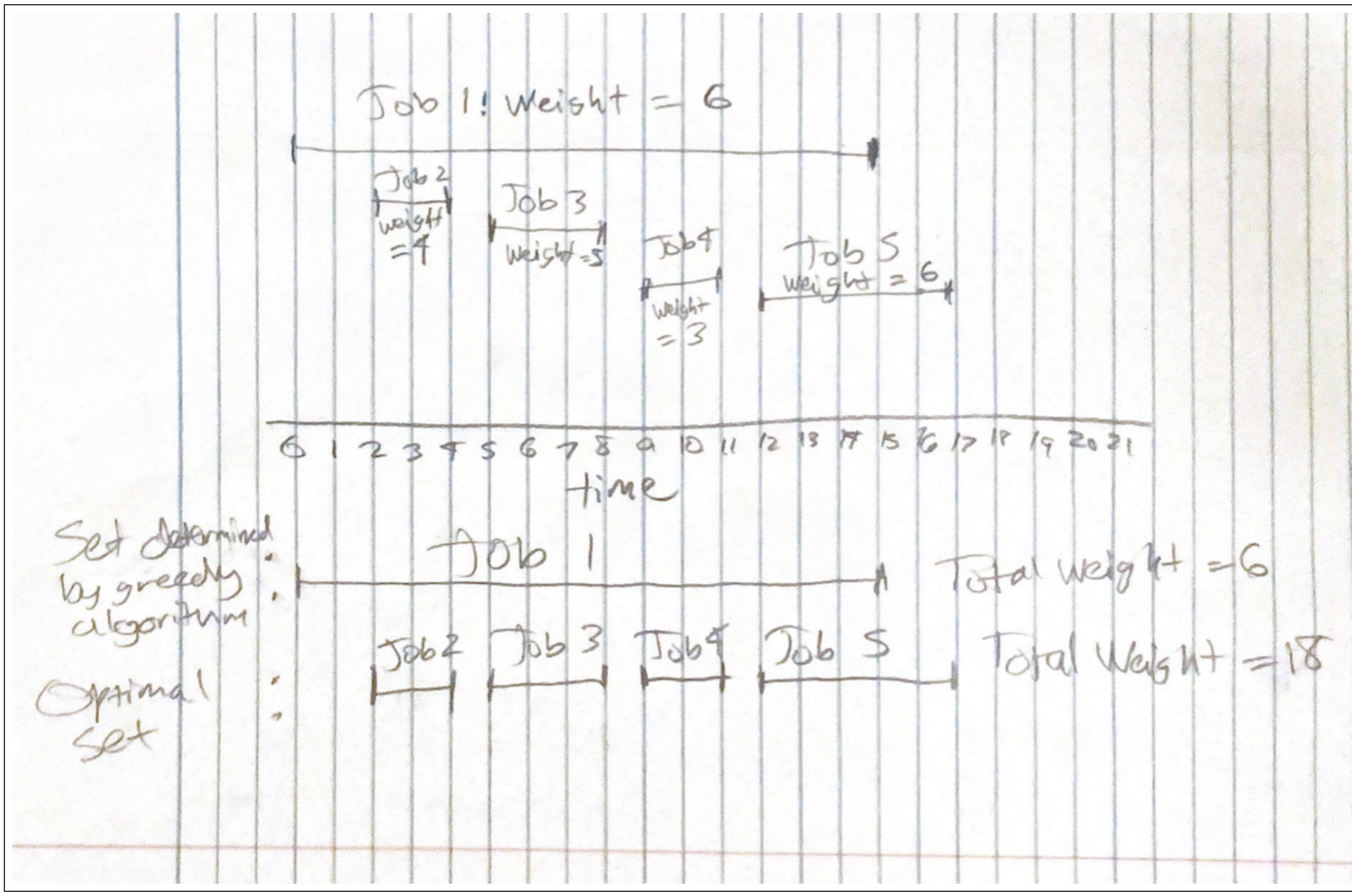
Problem 6. Consider now the Weighted Interval Scheduling problem, where each interval i is specified by

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

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.



□

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. Proof by exhaustion: Case $0 - 9$ cents: No dimes can be used as there's fewer than 10 cents

Case $10 - 19$ cents: One dime can be and is used in the optimal solution but no more will fit

Case $20 - 24$ cents: Two dimes can be and are used in the optimal solution but no more will fit

Case $25 - 34$ cents: No dimes are used in the optimal solution because a quarter is used and once the quarter is used there's less than ten cents left so a dime doesn't fit

Case $35 - 44$ cents: After a quarter is used, one dime is used then no more will fit

Case $45 - 50$ cents: After a quarter is used, two dimes are used then no more will fit

Case $50 - 59$ cents: No dimes are used in the optimal solution because two quarters are used and once the quarters are used there's less than ten cents left so a dime doesn't fit

Case $60 - 69$ cents: After two quarters are used, one dime is used then no more will fit

Case $70 - 74$ cents: After two quarters are used, two dimes are used then no more will fit

Case $75 - 84$ cents: No dimes are used in the optimal solution because three quarters are used and once the quarters are used there's less than ten cents left so a dime doesn't fit

Case $85 - 94$ cents: After three quarters are used, one dime is used then no more will fit

Case $95 - 99$ cents: After three quarters are used, two dimes are used then no more will fit

Since we can give no more than 99 cents worth of change in this problem all cases have been exhausted and no more than two dimes are used in any of them. □

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. $S = \{0.95, 1.8\}$ $\square$

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 $[s, f] \in \mathcal{I}$.

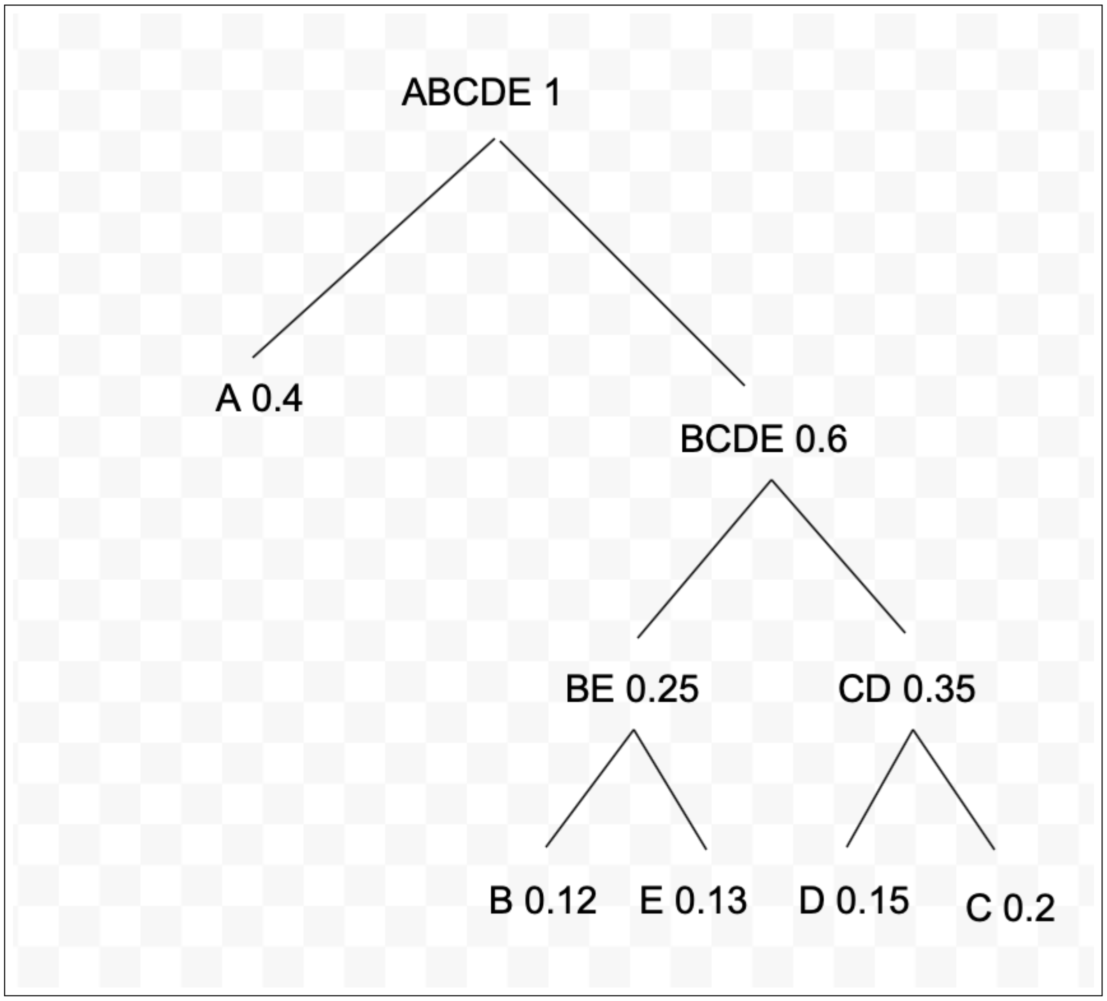
Proof. Assume S is a minimum projection set. If the points in S are all increased in value until they are the right endpoint of the interval whose right endpoint is closest, then the resulting set, S', would still have the same number of points as S and would still satisfy the requirements for a projection set since none of the points would have left any of the intervals they were in. Q. E. D. □

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.



Codes: A: 0 B: 100 C: 111 D: 110 E: 101 □