NAME: ____________________________

SPIRE ID: ____________________________

COMPSCI 250
Introduction to Computation
Second Midterm Fall 2025 – Solutions

D. A. M. Barrington and M. Golin 10 November 2025

DIRECTIONS:

- Answer the problems on the exam pages.
- There are four problems on pages 213, some with multiple parts, for 100 total points plus 5 extra credit. Final scale will be determined after the exam.
- We are also providing you with one blank piece or paper to use for scrap. Do NOT write final answers on this since we will not mark anything not in the exam booklet.
- If you need extra space use the back of a page – both sides are scanned.
- But, if you do write on the back, you must explicitly add a note on the front side stating that you are continuing on the back page. Otherwise, we might not see your solution on Gradescope.
- No books, notes, calculators, or collaboration.
- In case of a numerical answer, an arithmetic expression like "$2^{17} - 4$" need not be reduced to a single integer.
- Your answers must be LEGIBLE, and not cramped. Write only short paragraphs with space between paragraphs

1	/20+5
2	/ 20
3	/40
4	/20
Total	/100+5

Family Name: ______________________

Question 1: Induction 1

(A) (10 points)

For naturals n, let $S(n)$ be the sum $1 \cdot 2^0 + 2 \cdot 2^1 + 3 \cdot 2^2 + \ldots + (n+1) \cdot 2^n$.

Formally, $S(n) = \sum_{i=0}^{n} (i+1)2^i$.

Examples: $S(0) = 1 \cdot 2^0 = 1$, $S(1) = 1 \cdot 2^0 + 2 \cdot 2^1 = 5$, $S(2) = 1 \cdot 2^0 + 2 \cdot 2^1 + 3 \cdot 2^2 = 17$.

Prove *by induction*, that for *all* naturals n, $S(n) = n2^{n+1} + 1$.

i) First write your induction hypothesis in the box below. This should be in the form $P(x)$, where you *must* explicitly explain what x is and write an unambiguous statement of $P(x)$.

Solution
$P(n) : S(n) = n2^{n+1} + 1$ for all naturals n.

(ii) Next, write your base case(s) in the box below.

Solution
Base Case: $n = 0$. This is correct because $S(0) = 1 \cdot 2^0 = 1 = 0 \cdot 2^0 + 1$.

(iii) Finally, provide your induction step. This step will be marked on how clear and mathematically precise your proof is. Ambiguous explanations will have points deducted.
Be sure to clearly describe your induction goal, explicitly identify where the induction hypothesis is being usesd and justify every one of your manipulation steps.

If you run out of space, continue the proof on the back page (with a note stating that you are writing on the back).

Solution: For all natural n, we need to prove $P(n) \rightarrow P(n+1)$. To see this, note that:

$$\begin{aligned}
S(n+1) &= \sum_{i=0}^{n+1} (i+1)2^i && \text{(Definition)} \\
&= \left(\sum_{i=0}^{n} (i+1)2^i \right) + ((n+1)+1)2^{n+1} && \text{(algebraic manipulation)} \\
&= S(n) + (n+2)2^{n+2} && \text{(Definition)} \\
&= (n2^{n+1} + 1) + (n+2)2^{n+1} && \text{(IH)} \\
&= 2n2^{n+1} + 2 \cdot 2^{n+1} + 1 && \text{(algebraic manipulation)} \\
&= (n+1)2^{n+2} + 1 && \text{(algebraic manipulation).}
\end{aligned}$$

Marking Notes.

The mean for this problem was 9/10, the median was 10/10, and 67% got full marks.

IMPORTANT. What was being marked was not just lack of errors. A proof that had no explanation and just jumped from the assumption to the conclusion would not have any logical errors.

What was being marked was also how clearly justified each step was. So, a jump from one step to another that wasn't obvious would have points deducted for "missing justification".

1. One point was deducted for answers that ended with something like

$$\begin{aligned} &= S(n) + (n+2)2^{n+2} && (\textit{Definition}) \\ &= (n2^{n+1} + 1) + (n+2)2^{n+1} && (\textit{IH}) \\ &= (n+1)2^{n+2} + 1 && (\textit{algebraic manipulation}) \end{aligned}$$

i.e., that skipped a step justifying why the algebraic manipulation worked.

That answer above was correct, but it wasn't sufficient.

To see why, we actually had some students answer (making a mistake on the 2nd line)

$$\begin{aligned} S(n+1) &= \sum_{i=0}^{n+1} (i+1)2^i && (\textit{Definition}) \\ &= \left(\sum_{i=0}^{n} (i+1)2^i\right) + (n+1)2^{n+1} && (\textit{algebraic manipulation} - \textbf{incorrect}) \\ &= S(n) + (n+1)2^{n+1} && (\textit{Definition}) \\ &= (n2^{n+1} + 1) + (n+1)2^{n+1} && (\textit{IH}) \\ &= (n+1)2^{n+2} + 1 && (\textit{algebraic manipulation}). \end{aligned}$$

The reason it wasn't immediately obvious their answer was incorrect is that they didn't provide the extra line of algebraic manipulation. If they had, it would have caught the error.

It's not the job of the reader to verify steps like this in the proof. It's the job of the proof writer to provide enough justification so that the proof reader can immediately see that it's correct. We therefore deducted for insufficient justification even in the first case.

Family Name: ____________________

(B) (10 points) **Solution**

For naturals n, let A_n be defined by

$$A_0 = 0, \quad A_1 = 1 \quad \text{and for all } n \geq 1, \ A_{n+1} = 7A_n - 12A_{n-1}.$$

Examples: $A_2 = 7A_1 - 12A_0 = 7$ and $A_3 = 7A_2 - 12A_1 = 7 \cdot 7 - 12 \cdot 1 = 37$.

Prove *by induction*, that for *all* naturals n, $A_n = 4^n - 3^n$.

As a validity check note that $A_2 = 7 = 4^2 - 3^2$ and $A_3 = 37 = 64 - 27 = 4^3 - 3^3$.

i) First write your induction hypothesis in the box below. This should be in the form $P(x)$, where you *must* explicitly explain what x is and write an unambiguous statement of $P(x)$.

Solution:
$P(n) : A_n = 4^n - 3^n$ for all naturals n.

(ii) Next, write your base case(s) in the box below.

Solution:
Base Cases: $n = 0$ and $n = 1$.
This is correct because $A_0 = 0 = 4^0 - 3^0$ and $A_1 = 1 = 4^1 - 3^1$.

(iii) Finally, provide your induction step. This step will be marked on how clear and mathematically precise your proof is. Ambiguous explanations will have points deducted.
Be sure to clearly describe your induction goal, explicitly identify where the induction hypothesis is being usesd and justify every one of your manipulation steps.

If you run out of space, continue the proof on the back page (with a note stating that you are writing on the back).

Solution:

We use strong induction.

Assume that, for $n \geq 1$, $P(i)$ is true for all $i \leq n$.
We need to prove that this implies $P(n+1)$:

$$\begin{aligned}
A_{n+1} &= 7A_n - 12A_{n-1} && \text{(Definition)} \\
&= 7\left(4^n - 3^n\right) - 12\left(4^{n-1} - 3^{n-1}\right) && \text{(two uses of IH)} \\
&= 4^n(7-3) - 3^n(7-4) && \text{(algebraic manipulation)} \\
&= 4^{n+1} - 3^{n+1},
\end{aligned}$$

i.e., that $A_{n+1} = 4^{n+1} - 3^{n+1}$, i.e., $P(n+1)$.

Marking Notes

The mean for this problem was 8.2/10, the median was 9.5/10, and 40% got full marks.

IMPORTANT. What was being marked was not just lack of errors. A proof that had no explanation and just jumped from the assumption sto the conclusion would not have any logical errors.

What was being marked was also how clearly justified each stop was. So, a jump from one step to another that wasn't obvious would have points deducted for "missing justification".

1B1. Some students' answers exhibited the LHS-RHS error discussed in class, e.g., Lecture 23, pp 15-16. When using a LHS-RHS type proof it was necessary to explicitly note that fact and clearly label what is being assumed and what is being proven. Without that extra information, no one could understand what the proof was doing.

As an example, just writing an "=" on every line with a LHS and RHS without clearly explaining WHAT the LHS was and what the RHS was would b ethis error.

1B2. Since the recurrence calls both $n-1$ *and* $n-2$*, the base case had to have two values,* $n = 0, 1$.

1B3. A small anount (0.25) was deducted for proving correctness for A_{n+2} *rather than* A_{n+1} *if the IS did not explicitly state that it was being proven for all* $n \geq 0$*. Otherwise, ythe dfualy would be to assume that* n *starts with the largest known value, i.e.,* $n = 1$*, which means that the proof would be skiping the* A_2 *case.*

1B4. Some students jumped directly from the IH derived

$$7\left(4^n - 3^n\right) - 12\left(4^{n-1} - 3^{n-1}\right)$$

saying that it was equal to

$$4^{n+1} - 3^{n+1}$$

without providing any justification. While this was not "incorrect" it was not a proof, so points were deducted. A proof needs to justify its steps and this is a complicated enough leap that it had to be explained (and the explanation was a major part of the proof).

Family Name: ____________________

(C) Extra Credit (5 points) Recall the Fibonacci numbers F_n are defined on the naturals n by: $F_0 = 0$, $F_1 = 1$, and,

$$\forall n \geq 1,\ F_{n+1} = F_n + F_{n-1}. \qquad (1)$$

The first 11 F_n are shown below.

F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
0	1	1	2	3	5	8	13	21	34	55

In CICS 210 we learn about a data structure called AVL trees and find that the minumum number of nodes N_n in an AVL tree of height n satisfies $N_0 = 1$, $N_1 = 2$, and,

$$\forall n \geq 1,\ N_{n+1} = N_n + N_{n-1} + 1. \qquad (2)$$

The first 8 N_n are shown below.

N_0	N_1	N_2	N_3	N_4	N_5	N_6	N_7
1	2	4	7	12	20	33	54

Prove, *by induction* that for all naturals, $N_n = F_{n+3} - 1$.

i) First write your induction hypothesis in the box below. This should be in the form $P(x)$, where you *must* explicitly explain what x is and write an unambiguous statement of $P(x)$.

Solution:
$P(n) : N_n = F_{n+3} - 1$ for all naturals n.

(ii) Next, write your base case(s) in the box below.

Solution:
Base Cases: $n = 0$ and $n = 1$.
This is correct because $N_0 = 1 = 2 - 1 = F_3 - 1$ and $N_1 = 2 = 3 - 1 = F_4 - 1$.

(iii) Finally, provide your induction step. This step will be marked on how clear and mathematically precise your proof is. Ambiguous explanations will have points deducted.
Be sure to clearly describe your induction goal, explicitly identify where the induction hypothesis is being usesd and justify every one of your manipulation steps.

If you run out of space, continue the proof on the back page (with a note stating that you are writing on the back).

Solution: We use strong induction.

Assume that $P(i)$ is true for all $i \leq n$. We need to prove that this implies $P(n+1)$.:

$$\begin{aligned}
N_{n+1} &= N_n + N_{n-1} + 1 && \text{(Definition)} \\
&= (F_{n+3} - 1) + (F_{n+2} - 1) + 1 && \text{(two uses of IH)} \\
&= F_{n+3} + F_{n+2} - 1 && \text{(algebraic manipulation)} \\
&= F_{n+4} - 1 && \text{(Definition of Fibonacci numbers)} \\
&= F_{(n+1)+3} - 1
\end{aligned}$$

Marking Notes

The mean for this problem was 2/5, the median was 2.6/10, and 24% got full marks. (Many students elected not to solve this extra credit problem, or only attempt the first and second parts)

IMPORTANT. What was being marked was not just lack of errors. A proof that had no explanation and just jumped from the assumptions to the conclusion would not have any logical errors.

What was being marked was also how clearly justified each step was. So, a jump from one step to another that wasn't obvious would have points deducted for "missing justification".

1C1. Some students' answers exhibited the LHS-RHS error discussed in class, e.g., that shown in Lecture 23, pp 15-16. When using a LHS-RHS type proof it was necessary to explicitly note that fact and clearly label what is being assumed and what is being proven. Without that extra information, no one could understand what the proof is doing.

As an example, just writing an "=" on every line with a LHS and RHS without clearly explaining WHAT the LHS was and what the RHS was would reflect this error.

Since this exact issue was warned against in class, we deducted points if it occurred.

1C2. Since the recurrence calls both $n-1$ and $n-2$, the base case had to have two values. These were $n = 0, 1$. Solutions that only had one and not two base cases or different bases cases, were considered wrong.

1C3. Some students wrote something like this:

$$\begin{aligned} N_{n+1} &= N_n + N_{n-1} + 1 && (\textit{Definition}) \\ &= (F_{n+3} - 1) + (F_{n+2} - 1) + 1 && (\textit{two uses of IH}) \\ &= F_{n+3} + F_{n+2} - 1 && (\textit{algebraic manipulation}) \\ &= F_{(n+1)+3} - 1 \end{aligned}$$

This jumped directly from the IH derived

$$F_{n+3} + F_{n+2} - 1$$

to the lst line

$$F_{n+4} - 1$$

without providing any justification. While this was not "incorrect" it was not a proof, so a little bit was deducted. To not have the points deducted it was sufficient to just say that this was Eq (1) or the definition of Fibonacci.

Family Name: ____________________

Question 2 (20): Induction 2

A **rooted unary-ternary tree (RUTT)** is constructed as follows:

R0: It is either a single node, which is its root, or

R1: it is a new node, its root, which is connected to the roots of one other RUTT or

R2: it is a new node, its root, which is connected to the roots of three other RUTT's.

Furthermore,

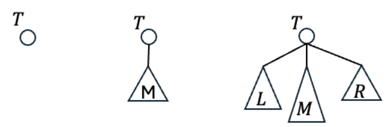
R3: The only RUTT's are those made by the first three rules.

Let $\mathcal{T}$ represent the set of all RUTT's. For tree $T \in \mathcal{T}$,

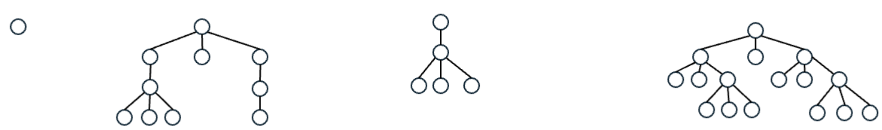
$$
\begin{aligned}
|T| &= \text{the number of nodes in } T,\\
L(T) &= \text{the number of leaves in } T,\\
I(T) &= \text{the number of internal nodes in } T,\\
d(T) &= \text{the depth of } T.
\end{aligned}
$$

Recall that the depth of T is the length of the longest path from the root to any leaf.

Diagrams (i-iii) are illustrations of the rules. Diagrams (iv)-(vii) are examples of some RUTT's and their associated values.



(i) (ii) (ii)



(iv)	(v)	(vi)	(vii)
$L(T) = 1$	$L(T) = 5$	$L(T) = 3$	$L(T) = 11$
$I(T) = 0$	$I(T) = 5$	$I(T) = 2$	$I(T) = 5$
$d(T) = 0$	$d(T) = 3$	$d(T) = 2$	$d(T) = 3$
$\lvert T\rvert = 1$	$\lvert T\rvert = 10$	$\lvert T\rvert = 5$	$\lvert T\rvert = 16$

Parts (A) and (B) of this problem are on the following pages. When writing the solutions to parts (A) and (B), you must use the the mathematical notation we provided above.

If your proof has multiple pieces, place each piece in a separate paragraph with space between the paragraphs. Be sure to clearly describe your induction goal.

If you run out of space, continue the proof on the back page of the problem (with a note stating that you are writing on the back).

(A, 10 points)

Prove, by induction, that for every $T \in \mathcal{T}$, it is true that $L(T)$ is odd.

a) First write your induction hypothesis in the box below. This should be in the form $P(x)$, where you *must* explicitly explain what x is and write an unambiguous statement of $P(x)$.

(b) Next, write your base case(s) in the box below.

(c) Finally, provide your induction step. This step will be marked on how clear and mathematically precise your proof is. Ambiguous explanations or explanations missing details will have points deducted.

Family Name: ____________________

(A, 10 points) **Solution 1, Structural Induction**

Prove, by induction, that for every $T \in \mathcal{T}$, it is true that $L(T)$ is odd.

a) First write your induction hypothesis in the box below. This should be in the form $P(x)$, where you *must* explicitly explain what x is and write an unambiguous statement of $P(x)$.

Solution. $P(T)$ is the statement: "$L(T)$ is odd".
$P(T)$ is defined over all trees $T \in \mathcal{T}$.

(b) Next, write your base case(s) in the box below.

Solution. The base case is $T = T_0$ where T_0 is the one node tree.
Since $L(T_0) = 1$ is odd $P(T_0)$ is correct

(c) Finally, provide your induction step. This step will be marked on how clear and mathematically precise your proof is. Ambiguous explanations or explanations missing details will have points deducted.

Solution:

The proof will be using structural induction. (Induction on size was also possible and is shown on the next page. Induction on depth was also possible.)

There are two cases, corresponding to T being built by R1 and R2.

- **T is built using R1, i.e., it is a root connected to the roots of one other RUTTs T_M:**

  By the IH, $L(T_M)$ is odd.

  When T is built using R_1, the leaves in T are exactly the same as the leaves in T_M so by the IH, $L(T) = L(T_M)$ is odd.

- **T is built using R2, i.e., it is a root connected to the roots of three other RUTTs T_L, T_M and T_R:**

  By the construction, $L(T) = L(T_L) + L(T_M) + L(T_R)$.

  By the IH, all of $L(T_L)$, $L(T_M)$ and $L(T_R)$ are odd.

  Since the sum of three odd numbers is odd, $L(T)$ is also odd.

Since R1 and R2 are the only ways of building a RUTT and we have shown that in both cases, $L(T)$ is odd, the proof is completed.

Family Name: ____________________

(A, 10 points) **Solution 2, Induction on size**

Prove, by induction, that for every $T \in \mathcal{T}$, it is true that $L(T)$ is odd.

a) First write your induction hypothesis in the box below. This should be in the form $P(x)$, where you *must* explicitly explain what x is and write an unambiguous statement of $P(x)$.

Solution. $P(n)$ is the statement: "If $T \in \mathcal{T}$ satisfies $|T| = n$. then $L(T)$ is odd".
$P(n)$ is defined over all positive naturals n.

(b) Next, write your base case(s) in the box below.

Solution. The base case is $n = 1$. $P(1)$ is true because 1 is odd.

(c) Finally, provide your induction step. This step will be marked on how clear and mathematically precise your proof is. Ambiguous explanations or explanations missing details will have points deducted.

Solution:

The proof will be by strong induction on n. Assume that $P(k)$ is true for all $k \leq n$..

To prove $P(n+1)$ let $T \in \mathcal{T}$ be an arbitrary RUTT satisfying $|T| = n+1$. We need to prove that $L(T)$ is odd. Since this is true for an arbitrary RUTT satisfying $|T| = n+1$ this will prove $P(n+1)$.

There are two cases, corresponding to T being built by R1 and R2.

- **T is built using R1, i.e., it is a root connected to the root of one other RUTTs T_M:**
  - When T is built using R_1, the leaves in T are exactly the same as the leaves in T_M so $L(T) = T(T_M)$.
  - By construction, $|T| = |T_M| + 1$, so $|T_M| = |T| - 1 = n + 1 - 1 = n$.
  - Since $|T_M| \leq n$, **By the IH,** $L(T_M)$ is odd.
  - Since $L(T) = L(T_M)$, $L(T)$ is also odd.
- **T is built using R2, i.e., it is a root connected to the roots of three other RUTTs T_L, T_M and T_R:**
  - By construction, $|T| = |T_L| + |T_M| + |T_R| + 1$, so $|T_L| \leq |T| - 1 = n$, $|T_M| \leq |T| - 1 = n$, and $|T_R| \leq |T| - 1 = n$.
    Therefore, by the induction hypothesis, $L(T_L)$, $L(T_M)$ and $L(L_R)$ are all odd
  - By the construction, $L(T) = L(T_L) + L(T_M) + L(T_R)$.
  - Since all $L(T_L)$, $L(T_M)$ and $L(T_R)$ are all odd and the sum of three odd items is odd, $L(T)$ is also odd.

Since R1 and R2 are the only ways of building a RUTT and we have shown that in both cases, $L(T)$ is odd and the proof is completed.

Marking Notes:

The mean for this problem was 7.9/10, the median was 10/10, and 53% got full marks.

This problem was very similar to a problem on the Spring 2024, Midterm 2, whose solution was discussed in great detail in the Lecture 23 slides.

IMPORTANT. What was being marked was not just lack of errors. A proof that had no explanation and just jumped from the assumptions to the conclusion would not have any logical errors.

What was being marked was also how clearly justified each step was. So, a jump from one step to another that wasn't obvious would have points deducted for "missing justification".

2A1. "Using Illegal Information" or "Expanding RUTT" Error.

The "Using Illegal Information" error is the one in which it is claimed that every RUTT is built by taking a leaf in some RTT and giving it one or three children, either leaving the number of leaves unchanged or increasing the number of leaves by 2.

This is considered a major error, It was explicitly discussed in the class notes and induction study/marking guide, where it was warned against. The ONLY way that you know to built an RUTT is via the rules given at the start of the problem. While it is true that every RUTT could be built "top down" by expanding a leaf in some other RUTT, that fact is a consequence *of R0, R1, R2 and would first need to be proven by induction before being used.*

This was described as "using Illegal Information" in the Induction Marking Guide *we distributed (Error F on page 12). Also see page 69 of the Lecture 23 slides where this exact error was described for the same problem on RBTs.*

The "Expanding RUTT Error" is one in which T_L, T_M and T_R are assumed to satisfy the IH and T is built out of them (rather than starting with T and defining *T_M or T_L, T_M and T_R as the one/three direct children of T. See page 10 of the* Induction Marking Guide.

2A2. Bottom-up Error.

This error is when the proof assumes $P(T)$ for some T and tries to use that T to build another T' and show that $P(T')$ is true. This is also considered a major error and was also explicitly discussed in the class notes and induction study/marking guide where it was warned against.

2A3. $P(T+1)$ Error or $P(T) \to P(T+1)$ error.

Some solutions were using structural induction on trees but then started discussing the "next" tree, $T+1$.

This was not possible. Trees are objects. "1" is a natural. There is no way to add a tree and a natural.

This error was discussed in detail in both the class notes and induction marking guide.

2A4. Did not define base case

Some solutions said the base case was T_0 with $L(T_0) = 1$ but did not define what T_0 was. This was a small point deduction. It would have been enough to say that T_0 was the one node tree or the tree from **R0**.

2A5. Missing **R1** *or* **R2** *case.*

This was usually (but not always) a solution which only proved the induction step for the **R2** *case but did not prove it for the* **R1** *case, i.e., when the RUTT is a root with one subtree. Since RUTT's can be built using either of the two cases, a correct induction must prove correctness of the IS for* **both** *cases.*

Family Name: ____________________

(B, 10 points)

Prove, *by induction,* that for every $T \in \mathcal{T}$, it is true that $|T| \leq \frac{3^{d(T)+1}-1}{2}$.

a) First write your induction hypothesis in the box below. This should be in the form $P(x)$, where you *must* explicitly explain what x is and write an unambiguous statement of $P(x)$.

(b) Next, write your base case(s) in the box below.

(c) Finally, provide your induction step. This step will be marked on how clear and mathematically precise your proof is. Ambiguous explanations or explanations missing details will have points deducted.

Family Name: ____________________

(B, 10 points) **Solution**

Prove, *by induction,* that for every $T \in \mathcal{T}$, it is true that $|T| \le \frac{3^{d(T)+1}-1}{2}$.

a) First write your induction hypothesis in the box below. This should be in the form $P(x)$, where you *must* explicitly explain what x is and write an unambiguous statement of $P(x)$.

Solution: $P(T)$ is the statement: "$|T| \le \frac{3^{d(T)+1}-1}{2}$".
$P(T)$ is defined over all trees $T \in \mathcal{T}$.

(b) Next, write your base case(s) in the box below.

Solution: The base case is $T = T_0$ where T_0 is the one node tree.
Since $d(T_0) = 0$ and $|T| = 1$, we have $|T_0| = 1 = \frac{3^1-1}{2} = \frac{3^{d(T)+1}-1}{2}$, so $P(T_0)$ is true.

(c) Finally, provide your induction step. This step will be marked on how clear and mathematically precise your proof is. Ambiguous explanations or explanations missing details will have points deducted.

Solution: The proof will be using structural induction. (Induction on depth or size was also possible.) There are two cases, corresponding to T being built by **R1** and by **R2**.

- **T is built using R2, i.e., it is a root connected to the roots of three other RUTTs T_L, T_M and T_R:**
  By the construction , $|T| = |T_L|+|T_M|+|T_R|+1$ and $d(T) = 1+\max(d(T_L), d(T_M), d(T_R))$. The last equation implies $d(T_i) \le d(T) - 1$ for all $i \in \{L, M, R\}$. So

$$\begin{aligned}
|T| &= |T_L| + |T_M| + |T_R| + 1 + 1 \\
&\le \frac{3^{d(T_L)+1}-1}{2} + \frac{3^{d(T_M)+1}-1}{2} + \frac{3^{d(T_R)+1}-1}{2} + 1 && \text{(three uses of IH)} \\
&\le \frac{3^{d(T)-1+1}-1}{2} + \frac{3^{d(T)-1+1}-1}{2} + \frac{3^{d(T)-1+1}-1}{2} + 1 && \text{(definition of } d(T)\text{)} \\
&= \frac{3 \cdot 3^{d(T)} - 3}{2} + 1 && \text{(algebraic manipulation)} \\
&= \frac{3^{d(T)+1}-1}{2} && \text{(algebraic manipulation).}
\end{aligned}$$

- **T is built using R1, i.e., it is a root connected to the roots of one other RUTT T_M:**
  By the IH, $L(T_M) \le \frac{3^{d(T_M)+1}-1}{2}$.
  When T is built using R_1, $|T| = |T_M| + 1$ and $d(T) = d(T_M) + 1$, so

$$\begin{aligned}
|T| &= |T_M| + 1 && \text{(Definition of size)} \\
&\le \frac{3^{d(T_M)+1}-1}{2} + 1 && \text{(IH)} \\
&= \frac{3^{d(T_M)+1}+1}{2} && \text{(algebraic manipulation)}
\end{aligned}$$

$$= \frac{3^{d(T)}+1}{2} \qquad \text{(Definition of depth)}$$

To complete the proof it is now **necessary** to show that

$$\frac{3^{d(T)}+1}{2} \le \frac{3^{d(T)+1}-1}{2}. \tag{3}$$

But, for any $x \ge 0$ we have $3^x + 2 \le 3^x + 2 \cdot 3^x = 3^{x+1}$ so $3^x + 1 \le 3^{x+1} - 1$ and

$$\frac{3^x+1}{2} \le \frac{3^{x+1}-1}{2}$$

proving Eq (3).

There are actually many ways of proving Eq (3). Here's another way that one student came up with on the exam.
Let T, T_M be as above and let T_0 be the RUTT with only one node from **R0***.*
Let T' be the RUTT built using **R2** *and $T_L = T_0 = T_R$ and the same T_M. Since $d(T_0) = 0$ we have*

$$d(T') = 1 + max(d(T_0), d(T_M), d(T_0) = 1 + dT(M) = d(T).$$

Now we can bootstrap of our proof of case **R2** *proven above to get*

$$\begin{aligned}
|T| &= 1 + |T_M| && \textit{(Definition of size)}\\
&\le 3 + |T_M| && \textit{(algebraic manipulation)}\\
&= |T'| && \textit{(Definition of size)}\\
&\le \frac{3^{d(T')}-1}{2} && \textit{(same proof as in R2)}\\
&= \frac{3^{d(T)}-1}{2} && \textit{(because } d(T) = d(T')\textit{).}
\end{aligned}$$

Since R1 and R2 are the only ways of building a RUTT and we have shown that in both cases, $|T| \le \frac{3^{d(T)+1}-1}{2}$, the proof is completed.

Marking Notes:

The mean for this problem was 6.53/10, the median was 7/10, and 20% got full marks.

This problem was very similar to a problem on the Spring 2024, Midterm 2 whose solution was discussed in great detail in the Lecture 23 slides.

IMPORTANT. What was being marked was not just lack of errors. A proof that had no explanation and just jumped from the assumptions to the conclusion would not have any logical errors.

What was being marked was also how clearly justified each step was. So, a jump from one step to another that wasn't obvious would have points deducted for "missing justification".

2B1. "Using Illegal Information" or "Expanding RUTT" Error.

This is the essentially same error as was described in the Marking Notes for part A. Please see detailed description there.

A similar error more specific to this case appeared in some solutions which wrote that increasing depth by 1 could only increase the maximum number of nodes in a RUTT by 3^d*. Again, while this is a correct statement, it would first have to be proven,* by induction *before it could be used.*

We emphasize that the ONLY information you know about RUTTs is **R0**, **R1** *and* **R2**. *ANY other fact, e.g, that the maximum number of nodes on level* i *of a RUTT is* 3^i*, would have to be PROVEN (by induction) before using it.*

An error similar to the one above was one in which it was stated that the RUTT with the maximum *number of nodes. i.e., worst-case, for a particular depth is the one in which every node on depth* $< d$ *has exactly three children and all leaves are at the same depth and then inverting the resulting equation to get the desired result. Again, while these are correct statements, they would first have to be proven,* formally by induction *before they could be used. This proof would not fit into the structure that was required for this problem.*

2B2. Bottom-up Error.

This is the exact same error as was described in the Marking Notes for part A. Please see detailed description there.

2B3. $P(T+1)$ *Error or* $P(T) \to P(T+1)$ *error.*

This is the exact same error as was described in the Marking Notes for part A. Please see description there.

2B4. $|T| = |T_L| + |T_M| + |T_R|$ *error.*

A small number of solutions wrote

$$|T| = |T_L| + |T_M| + |T_R|$$

instead of

$$|T| = 1 + |T_L| + |T_M| + |T_R|.$$

While this might seem like a trivial error this means that the resulting equation derived by the induction would not be correct. If the +1 was missing, then the best inequality the resulting induction could derive would be

$$|T| \leq 3^{d(T)}$$

The error being marked here is not the typo, but not catching the later induction conclusion error.

2B5. Missing **R1** *or* **R2** *case.*

This was usually a solution which only proved the induction step for the **R2** *case but did not prove it for the* **R1** *case, i.e., when the RUTT is a root with one subtree. Since RUTT's can be built using either of the two cases, a correct induction must prove correctness of the IS for both cases.*

While it might seem that the **R1** *proof is a special case of the* **R2** *proof with two of the subtrees being empty, this is not formally correct. The RUTT definition does not permit empty subtrees, so the IH cannot be applied to empty subtrees.*

This is why the alternative proof for **R1** *given above could not just assume that* T_L *and* T_R *were empty. It was why we had to create the new* T' *and show that it had the same depth as* T *and work from there.*

Question 3 (40):

In this problem, we are going to carry out four searches on three graphs: an undirected graph U, a directed graph D, and an undirected labeled graph G. Each graph has the same six nodes, named for planets in the *Star Trek* universe. Each graph has the same nine edges, but they have the edge types appropriate to their type of graph.

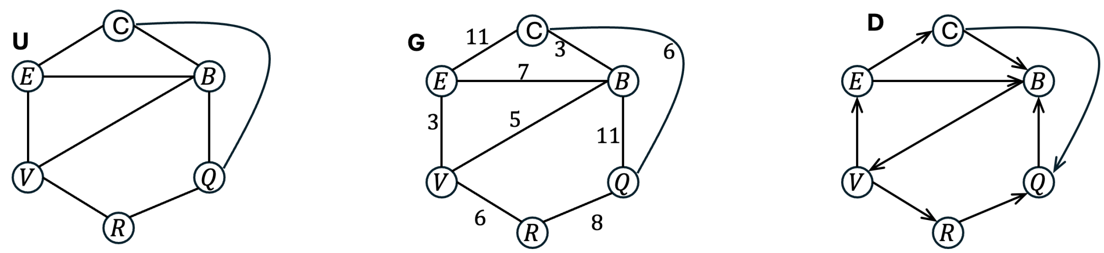
Because of technicalities involving faster-than-light travel, and navigation hazards such as black holes and quasars, it is not always possible to travel *directly* from any planet to any other planet.

Graph U has an edge between any two nodes (x, y) that allow direct travel from x to y or vice versa.

Graph D describes a scenario in which (due to interstellar treaties) each of the edges in U may only be traversed in one direction.

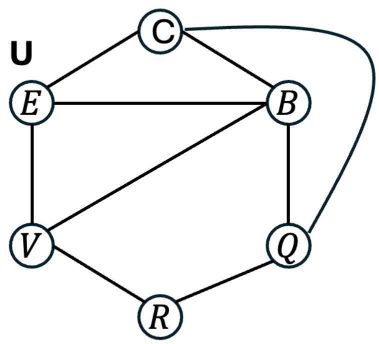
Finally, each edge (x, y) of graph G is labeled with the number of days a starship takes to travel from x to y, or from y to x.

The planets in the graph are Bajor (B), Cardassia Prime (C), Earth (E), Qo'noS (Q), Romulus (R), and Vulcan (V).



- (A, 10) **Undirected Breadth-First Search:**

  (i) Carry out a **BFS search** for the **undirected** graph U, starting with Earth and with **no goal node**.

  List all the events that occur in the running of the BFS, in the order in which they occur.

  An event is either the placement of a list-item on the open list or the removal of a list-item off the open list. Describe list-items in the form of pairs such as "(a, b)", meaning that the node a went on the list while node b was processed.

  When two or more list-items need to come off the open list, and they entered at the same time, take the one first that comes earlier alphabetically.

  Solution to (A, i):
  Here is the order of events:

  - $(E, --)$ begins on the queue.
  - $(E, --)$ comes off, (B, E), (C, E) and (V, E) go on.
  - (B, E) comes off, (C, B), (Q, B), and (V, B) go on.
  - (C, E) comes off, (Q, C) goes on.
  - (V, E) comes off, (R, V) goes on.
  - (C, B) is noted as a non-tree edge.
  - (Q, B) comes off, (R, Q) goes on.
  - (V, B) is noted as a non-tree edge.
  - (Q, C) is noted as a non-tree edge.
  - (R, V) comes off, nothing goes on.
  - (R, Q) is noted as a non-tree edge.

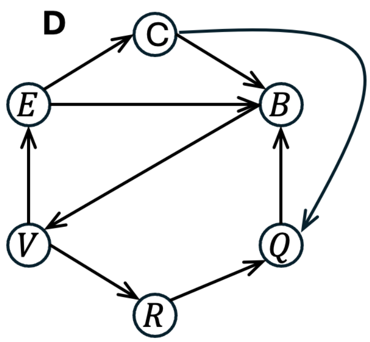
  (ii) In addition to writing the list, **also draw** the BFS tree from this search, starting with Earth and with no goal node. Indicate the tree edges and the non-tree edges.

  Solution to (A, ii): Node E is the root. It has parents B, C, and V. Q is a parent of B and R is a parent of V. There are non-tree edges (C, B), (V, B), (Q, C) and (Q, R).

  Grading Notes for Part (A): The mean was 8.6/10, the median was 10/10, and 53% got full marks. The most common mistakes were missing a step, or interpreting a correct sequence of events as a tree.

- (B, 10) **Directed Depth-First Search:**

  (i) Carry out a **DFS search** for the **directed** graph D, starting with Earth and **with no goal node**.



List all the events that occur in the running of the DFS, in the order in which they occur. An event is either the placement of a list-item on the open list or the removal of a list-item off the open list.

Describe list-items in the form of pairs such as "(a, b)", meaning that the node a went on the list while node b was processed.

When two or more nodes need to come off the open list, and they entered at the same time, take the one first that comes earlier alphabetically.

Solution to (B, i):

- $(E, --)$ begins on the stack.
- $(E, --)$ comes off, (C, E) and (B, E) go on.
- (B, E) comes off, (V, B) goes on.
- (V, B) comes off, (E, V) is identified as a back edge, and (R, V) goes on.
- (R, V) comes off, (Q, R) goes on.
- (Q, R) comes off, (B, Q) is identified as a back edge, and nothing goes on.
- (C, E) finally comes off, and (B, C) and (Q, C) are identified as cross edges.

(ii) In addition to writing the list, **also draw** the DFS tree for this search, indicating the tree and non-tree edges, starting with Earth and with no goal node. For each non-tree edge, **identify** it as a back, cross, or forward edge.

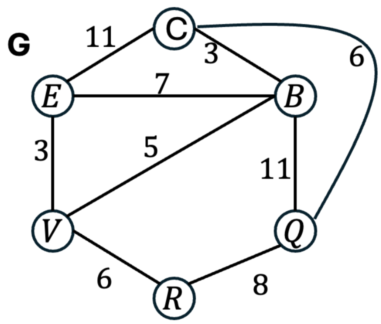
Solution to (B, ii): E is the root, with children B and C. B has child V, V has child R, and R has child Q. Q and C are leaves. There are back edges from V to E and from Q to B. There are cross edges from C to B and to Q.

Grading Notes for Part B:

The mean was 8.2/10, the median was 8/10, and 33% got full marks. The biggest problem students has was to identify the types of the non-tree edges. Some also got the tree structure wrong, for example by reversing the initial visits to B and C. A lot of 9/10 scores resulted from correct trees where too many or too few node visits occurred during the sequence.

- (C, 10) **Uniform-Cost Search:**

  (C) Conduct a uniform-cost search of the labeled undirected graph G, with Earth as the start node and Qo'noS as the goal node.



List all the events that occur in the running of the UCS, in the order in which they occur. An event is either the placement of a list-item on the open list or the removal of a list-item off the open list.

Describe list-items using triples such as "(a, t, b)", meaning that this node a went on the list while b was processed, with t being its priority value.

When two or more list-items need to come off the list, they should follow the rules of uniform-cost, and you may break ties as you like.

Solution: Here is the correct order of events:

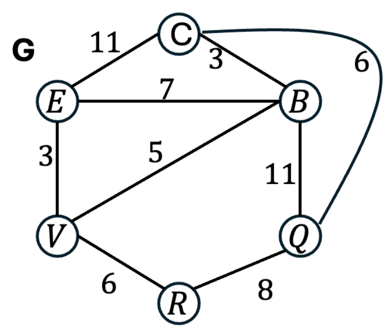
- $(E, 0, --)$ begins on the priority queue.
- $(E, 0, --)$ comes off, $(B, 7, E)$, $(C, 11, E)$, and $(V, 3, E)$ go on.
- $(V, 3, E)$ comes off, $(B, 8, V)$ and $(R, 9, V)$ go on.
- $(B, 7, E)$ comes off, $(C, 10, B)$ and $(Q, 18, B)$ go on.
- $(B, 8, V)$ is discarded as B is on the closed list.
- $(R, 9, V)$ comes off, $(Q, 17, R)$ goes on.
- $(C, 10, B)$ comes off, $(Q, 16, C)$ goes on.
- $(C, 11, E)$ is discarded as C is on the closed list.
- $(Q, 16, C)$ comes off and we declare victory.
- $(Q, 17, R)$ and $(Q, 18, B)$ remain on the queue at the end.
- We have found the optimal path of length 16 from E to Q, using the path E-B-C-Q.

Grading Notes for Part (C): The mean was 8.1/10, the median 10/10, and 54% got full marks. I charged a three-point penalty for those (30% of you) that did not get the correct answer for the distance to Qo'noS. The most common way to get this wrong was to miss the node $(C, 10, B)$, either by not seeing it when you processed $(B, 7, E)$ or not processing it properly. It was important that the node $(Q, 17, R)$ be put on the PQ, but not taken off (unless you continue the process after declaring victory).

- (D, 10) A^* **Search:**

We will now ask you to conduct an A^* search for the labeled undirected graph G using the following heuristic function h. For any node x, $h(x)$ will be the a lower bound on the distance in G from x to Q. Here are the values of h:

$$\begin{array}{rclrclrcl} h(B) &=& 7, & h(C) &=& 4, & h(E) &=& 12 \\ h(Q) &=& 0, & h(R) &=& 8, & h(V) &=& 12 \end{array}$$



Conduct an A^* search of the labeled graph G to determine the shortest path and the distance along that from Earth to Qo'noS, using the heuristic function h above. (Thus Earth is the start node and Qo'noS is the goal node.)

List all the events that occur in the running of the A^* search, in the order in which they occur. An event is either the placement of a list-item on the open list or the removal of a list-item off the open list.

Describe list-items using tuples such as "(a, t, d, b)", meaning that this node a went on the list while b was processed, with t being its priority value and d being the distance (along the known path) from Earth to a.

When two or more list-items need to come off the list, they should follow the rules of uniform-cost, and you may break ties as you like.

Solution: Here is the correct order of events:

- $(E, 12, 0, --)$ begins on the priority queue.
- $(E, 12, 0, --)$ comes off, $(B, 14, 7, E)$, $(C, 15, 11, E)$, and $(V, 15, 3, E)$ go on.
- $(B, 14, 7, E)$ comes off, $(C, 14, 10, B)$, $(V, 24, 12, B)$ and $(Q, 18, 18, B)$ go on.
- $(C, 14, 10, B)$ comes off, $(Q, 16, 16, C)$ goes on.
- $(C, 15, 11, E)$ is discarded as C is on the closed list.
- $(V, 15, 3, E)$ comes off, $(R, 17, 9, V)$ goes on.
- $(Q, 16, C)$ comes off and we declare victory.
- $(R, 17, 9, V)$, $(Q, 17, 17, R)$ and $(Q, 18, 18, B)$ remain on the queue at the end.
- We have found the optimal path of length 16 from E to Q, using the path E-B-C-Q.
- The heuristic allowed us to skip exploring node R.

Grading Notes for Part (D): The mean was 7.9/10, the median 9/10, and 43% got full marks. Again I took off three points for getting the final answer of 16 incorrect, or not stating it clearly. Again the problem node was $(C, 14, 10, B)$, either not finding it from $(B, 14, 7, B)$ or not correctly processing it. (A surprising number of people added $10 + 6 = 17$ and thus got the wrong length for the correct path, or rejected it in favor of the legitmate path of length 17.) The "runaway number" problem was less common this time, but still occurred, where people were considering paths that revisited nodes, or double-counted heuristics. As with part (C), many of the 9/10 scores were for correct paths that involved too few or too many nodes explored in the sequence. (For example, $(V, 15, 3, E)$ has to be explored before we declare victory, and thus $(R, 17, 9, V)$ is put on the PQ to be left there at the end.)

Question 4 (20) The following are ten true/false questions, with no explanation needed or wanted, no partial credit for wrong answers, and no penalty for guessing.

After reading the questions, write the correct answer, either T (for true) or F (for false), in the corresponding column.

(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)
T	F	F	F	T	T	F	F	F	T

The mean score was 13.09/20 and the median 14/20, with 4% getting perfect scores.

(a) Let $P(n)$ be a predicate on the naturals. If $\forall n : P(n)$ can be proved by ordinary induction, then it can also be proved by strong induction.

TRUE (92% correct). Any inductive step $P(n) \rightarrow P(n+1)$ also proves $Q(n) \rightarrow P(n+1)$.

(b) Let $P(n)$ be a predicate on the naturals, and assume that $P(1)$ and $\forall n : P(n) \rightarrow P(n+2)$ are true. Then $P(n)$ must be true for all positive naturals.

FALSE (80% correct). We know that $P(n)$ is true for all odd naturals, but we don't know whether $P(2)$, for example, is true.

(c) Let $\Sigma = \{a\}$ be an alphabet with one letter, and consider the set of strings Σ^*. Define a function f from $\mathbb{N}$ to Σ^* so that $f(n) = a^n$ and a function g from Σ^* to $\mathbb{N}$ such that $g(w) = |w|$. Then f and g are not inverses of one another.

FALSE (62% correct). They are.

(d) Consider an infinite plane crossed by n (infinite) lines, no two of which are parallel. Then the lines divide the plane into exactly $\frac{n^2+n+2}{2}$ regions.

FALSE (43% correct). This is true if no three lines intersect in a common point, as we proved in lecture, but without that assumption there could be fewer regions.

(e) Let T be a tree with four or more nodes. Then it may be possible to add two edges to T to form a new graph G that has three different cycles.

TRUE (72% correct). For an example, let T have edges (a, b), (b, c), and (a, d), and let G also contain the edges (a, d) and (a, c). There are cycles in G a-c-d, a-b-c, and a-b-c-d.

(f) Let F be a forest consisting of exactly two trees T and U. If we create a new graph G by adding one edge to F, with one endpoint in T and the other in U, then G must be a tree.

TRUE (69% correct). There can be no cycles that don't use the new edge. But any path using that edge must start in T and end in U, or vice versa, so it cannot revisit its starting node.

(g) Let G be a graph containing nodes u and v, and consider both a BFS and a DFS of G, with start node u and goal node v, where each search uses a closed list to avoid revisiting nodes. Then it is possible that the BFS reaches v and the DFS does not.

FALSE (68% correct). With these conditions, each search will succeed if and only any path from u to v exists. We probably should have added that the graph was finite—we said that it was when anyone asked. Every graph we have worked with in the course has been finite.

(h) Consider a BFS (using a closed list) of an undirected graph that is a forest. Then it is possible for two copies of the same node to be stored on the queue at the same time during the search.

FALSE (63% correct). We only get two copies because there are two different paths from the start node to the same node. But this cannot happen in a forest due to the Unique Path Theorem.

(i) Let G be a weighted graph with positive edge costs, let g be a node of G, and let $h(x)$ be an admissible, consistent heuristic for an A^* search where g is the goal node. Let s and y be two nodes of G such that the best-path distance from s to y is less than the best-path distance from s to g. Then, as in the corresponding uniform cost search, an A^* search from s to g will eventually take y off of its priority queue before taking g off.

FALSE (44% correct). The whole point of A^* is that the heuristic might give y a higher priority than g in the priority queue, so that the search might succeed before y is explored.

(j) Consider any two-player game represented by a finite game tree such that there is more than one leaf, each leaf value is either $+1$ or -1, and each internal node has exactly two children. If the first player to move has a winning strategy, then it is possible to change at least one of the leaf values to its opposite without changing the value of the game.

TRUE (63% correct). Let x and y be the children of the root r. Without loss of generality, let r be a White node, so that $v(r) = +1 = \max(v(x), v(y))$. If $v(x) = +1$, then $v(r) = +1$ no matter what $v(y)$ is, so we can change a leaf value in y's tree. If $v(y) = +1$, by the same argument we can change a leaf value in x's tree.