NAME: _____

SPIRE ID: _____

COMPSCI 250
Introduction to Computation
Second Midterm Spring 2025 – Solution Key

D. A. M. Barrington and M. Golin 14 April 2025

DIRECTIONS:

- Answer the problems on the exam pages.
- There are four problems on pages 2-12, some with multiple parts, for 100 total points plus 5 extra credit. Final scale will be determined after the exam.
- Page 13 contains useful definitions and is given to you separately – do not put answers on it!
- 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:

(A) (10 points)

"According to legend, a king in India was so pleased with the new game of chess, that he offered the inventor anything he asked. The inventor asked for a few grains of rice to be placed on a chessboard for him to take home. More specifically, he asked for one grain of rice to be placed on the first square of the board, two on the second, four on the third, and so on. Thus 2^{i-1} grains would be placed on the i'th square.

Let $S(n)$ be the number of grains of rice that the inventor receives after rice is placed on n squares.

Prove, by ordinary induction on all positive naturals, that $S_n = 2^n - 1$.

Please note that $2^n - 1$ and 2^{n-1} are in general different numbers." Also, your proof MUST be by induction. You may not use the formula for geometric series.

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: $P(n)$ is "$S_n = 2^n - 1$". This is defined for all naturals $n \geq 1$.

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

Solution: Base case is $P(1)$. This is true because $S_1 = 1 = 2^1 - 1$..

(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: Assume $P(n)$. Goal is to prove $P(n+1)$, i.e., $S_{n+1} = 2^{n+1} - 1$,

$$\begin{aligned}
S_{n+1} &= \sum_{i=1}^{n+1} 2^{i-1} && \text{(Definition of } S_n\text{)} \\
&= \sum_{i=1}^{n} 2^{i-1} + 2^{(n+1)-1} && \\
&= S_n + 2^{(n+1)-1} && \text{(Definition of } S_n\text{)} \\
&= (2^n - 1) + 2^{(n+1)-1} && \text{(IH)} \\
&= 2^n + 2^n - 1 && \text{(Algebraic Manipulation)} \\
&= 2^{n+1} - 1. && \text{(Algebraic Manipulation)}
\end{aligned}$$

Marking Notes: What was being marked here was not if you knew the answer. What was being marked was how clear and correct the proof was. Would someone who did not already know the answer be able to follow the logic in your proof and verify that it is correct? If not, the solution could not receive full marks.

Some common errors.

1. **After defining $P()$ properly as the predicate "$S_n = 2^n - 1$", proof starts with $P(n+1) = \sum_{i=1}^{n+1} 2^n$ or $P(n+1) = P(n) + 2^n$, confusing $P(n)$ and $S(n)$.**
2. **Stating that $S(n) \rightarrow S(n+1)$.**
   This makes no sense since $S(n)$ (S_n) is a number, not a predicate.
3. **The Base Case was $n = 1$. The problem started with $n = 1$. S_0 was not defined. So $n = 0$ would not be a correct base case.**
4. **The problem instructions required that the solution must explicitly identify where the IH is used (similar to many of the proofs done in class). Writing the IH at the top of the IS and then using it later without identifying where it was being used did not satisfy this requirement.**
5. **A correct inductive proof must, somewhere inside, have explicitly written $S_{n+1} = S_n + 2^n$ or something equivalent.**

Family Name: _____

(B) (10 points)

Recall the Fibonacci numbers F_n defined on the naturals n by: $F_0 = 1$, $F_1 = 1$, and,

$$\forall n \geq 1,\ F_{n+1} = F_n + F_{n-1}. \tag{1}$$

F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
0	1	1	2	3	5	8	13

Let $S_n = \sum_{i=0}^{n} F_i$ be the sum of the first $n+1$ Fibonacci numbers:
Example: $S_5 = 0 + 1 + 1 + 2 + 3 + 5 = 12$. Note that $S_5 = F_7 - 1$.

Prove, *using induction*, that, for all positive naturals n, $S_n = F_{n+2} - 1$.

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: $P(x)$ is "$S_n = F_{n+2} - 1$". This is defined for all naturals.

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

Solution: Base case is $P(1)$. This is true because $S_1 = 0 + 1 = 2 - 1 = F_3 - 1$.

(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: Assume $P(n)$. Goal is to prove $P(n+1)$, i.e., $S_{n+1} = F_{n+3} - 1$.

$$\begin{aligned}
S_{n+1} &= \sum_{i=0}^{n+1} F_i && \text{(Definition)} \\
&= \sum_{i=0}^{n} F_i + F_{n+1} && \text{(Manipulation)} \\
&= S(n) + F_{n+1} && \text{(Definition)} \\
&= F_{n+2} - 1 + F_{n+1} && \text{(IH)} \\
&= F_{n+3} - 1 && \text{(Fact that } F_{n+3} = F_{n+2} + F_{n+1} \text{ from Eq. (1))}
\end{aligned}$$

Marking Notes: What was being marked here was not if you knew the answer. What was being marked was how clear and correct the proof was. Would someone who did not already know the answer be able to follow the logic in your proof and verify that it is correct? If not, the solution could not receive full marks.

1. **Proof needed to somewhere use the fact that $S(n+1) = S(n) + F_{n+1}$.**
2. **Points were deducted for solutions that equated a LHS with a RHS without clearly specifying what was the goal and what was being assumed.**

   As similar error was writing a series of "=" equations with calculations on different sides but not explicitly labelling what each side was, i.e., a valid statement (from assumption) or goal. Without explicit labelling providing the connective tissue, a series of statements is NOT a clear unambiguious proof.

(C) Extra Credit (5 points)

For natural n, define $_n$ by $A_0 = 2$, $A_1 = 1$ and

$$\forall n \geq 1,\ A_{n+1} = A_n + 6A_{n-1}. \tag{2}$$

As examples, $A_2 = 1 + 6 \cdot 2 = 13$ and $A_3 = 13 + 6 \cdot 1 = 19$.

Prove, *using induction*, that $A_n = 3^n + (-2)^n$.

As a sanity check, note that $A_3 = 19 = 27 - 8 = 3^3 + (-2)^3$.

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: $P(x)$ is "$A_n = 3^n + (-2)^n$". This is defined for all naturals.

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

Solution: Base cases are $P(0)$ and $P(1)$. $A_0 = 2 = 3^0 + (-2)^0$. $A_1 = 1 = 3^1 + (-2)^1$.
So $P(0)$ and $P(1)$ are true.

(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: This needs strong induction Let $n \geq 1$. Assume $P(i)$ is true for all $i \leq n$. Then

$$\begin{aligned}
A_{n+1} &= A_n + 6A_{n-1} \\
&= (3^n + (-2)^n) + 6(3^{n-1} + (-2)^{n-1}) \qquad \text{(Two uses of the IH)} \\
&= \left(3^n + 6 \cdot 3^{n-1}\right) + \left((-2)^n) + 6 \cdot (-2)^{n-1}\right) \\
&= (3+6)\,3^{n-1} + (-2+6)\,(-2)^{n-1} \\
&= 9 \cdot 3^{n-1} + 4 \cdot (-2)^{n-1} \\
&= 3^{n+1} + (-2)^{n+1}
\end{aligned}$$

Marking Notes. Some common errors were:

1. **Only having one base case. This problem required two base cases, and they had to be $n = 0, 1$.**

   The problem explicitly said that A_{n+1} is only defined by Equation (2) for $n \geq 1$. This means that $P(0)$ and $P(1)$ need to be checked without using the equation.

2. **Stating "$P(n) \to P(n+1)$" would be an error.**
   The proof actually needed $(P(n-1) \wedge P(n)) \to P(n+1)$, e.g., strong induction.

Family Name: _____

Question 2 (20): Induction 2

A **rooted ternary tree (RTT)** 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 exactly three other RTT's.

Furthermore,

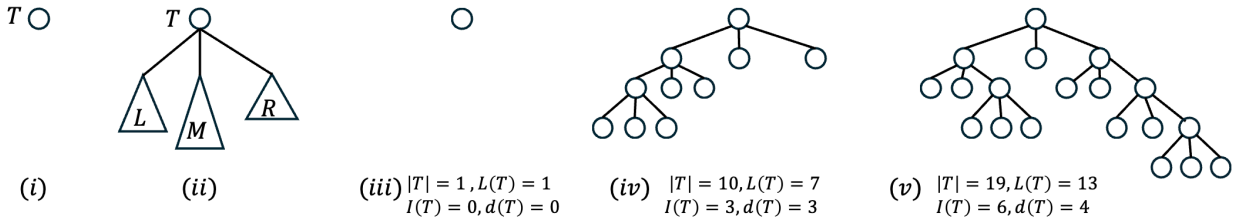
R2: The only RTT's are those made by the first two rules.

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

$\lvert T\rvert$	=	the number of nodes in T,	$L(T)$	=	the number of leaves in T,
$I(T)$	=	the number of internal nodes in T,	$d(T)$	=	the depth of T.

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

Diagrams (i-ii) are illustrations of the rules. Diagrams (iii), (iv) and (v) are examples of some RTT's and their associated values.



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).

Family Name: _____

(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.

Solution. Here is a proof using structural induction. (Next solution to same problem will use induction by size.)

(a)

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

(b)

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

(c) **Solution.** Since $T \neq T_0$ we have that T is a root connected to the roots of 3 other RTTs, L, M and R.

- $L(T) = L(L) + L(M) + L(R)$.
- By the Induction Hypothesis we know $P(L)$, $P(M)$ and $P(R)$. Thus all of $L(L)$, $L(M)$ and $L(R)$ are odd.
- Since $L(T)$ is the sum of three odd numbers, $L(T)$ is odd.

Second Solution. Here is a proof using strong induction on size. (Previous solution to same problem was via structural induction.).

(a)

Solution. $P(n)$ is the statement: "If $T \in \mathcal{T}$ has size $|T| = n$, then $L(T)$ is odd".
n is any positive natural.

(b)

Solution. The base case is $n = 1$. Let T_0 be the one node tree.
This is the only tree with size 1.
Since $L(T_0) = 1$, $P(1)$ is correct

(c) **Solution.**

Let $n > 1$. Assume that $P(i)$ is true for all $i \leq n$.

Let T be any RTT with $|T| = n + 1$.

Thus,

- T is a root connected to the roots of 3 other RTTs, L, M and R.
- $L(T) = L(L) + L(M) + L(R)$.
- Since $|L|, |M|, |R|$ are all less than $n + 1 - 3 = n = 2$, we know that $P(|L|)$, $P(|M|)$, and $P(|M|)$ are all true by the IH (strong induction).
- Thus all of $L(L)$, $L(M)$ and $L(R)$ are odd.
- Since $L(T)$ is the sum of three odd numbers, $L(T)$ is odd.

We have just seen that $L(T)$ is odd for every $T \in \mathcal{T}$ with $|T| = n + 1$. So $P(n + 1)$ is true.

Marking Notes:

1. "Using Illegal Information" or Expanding RTT" Error.

The "Using Illegal Information" error is the one in which it is claimed that every RTT is built by taking a leaf in some RTT and giving it three children, 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 RTT is via the rules given in the exam. While it is true that every RTT could be built "top down" by expanding a leaf in some other RTT, 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 RTT 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_L, T_M and T_R as the three direct children of T. See page 10 of the *Induction Marking Guide*.

2. 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.

3. $P(T+1)$ 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.

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.

Solution. Here is a proof using structural induction. (Next solutions to same problem will use induction by size and then depth.)

(a)

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

(b)

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

(c) **Solution.**

Since $T \neq T_0$ we have that T is a root connected to the roots of 3 other RTTs, L, M and R.

- $|T| = 1 + |L| + |M| + |R|$
- $d(T) = 1 + \max(d(L), d(M), d(R))$.
- By the Induction Hypothesis we know $P(L)$, $P(M)$ and $P(R)$. Thus,

$$|L| \leq \frac{3^{d(L)+1}-1}{2}, \quad |M| \leq \frac{3^{d(M)+1}-1}{2} \quad \text{and} \quad |R| \leq \frac{3^{d(R)+1}-1}{2}$$

- Since $d(L)+1 \leq d(T)$, $d(M)+1 \leq d(T)$, and $d(R)+1 \leq d(T)$, this implies

$$|L| \leq \frac{3^{d(T)}-1}{2}, \quad |M| \leq \frac{3^{d(T)}-1}{2} \quad \text{and} \quad |R| \leq \frac{3^{d(T)}-1}{2}.$$

Combining the pieces above gives

$$\begin{aligned} |T| &= 1 + |L| + |M| + |R| \\ &\leq 1 + 3\left(\frac{3^{d(T)}-1}{2}\right) \\ &= \frac{3^{d(T)+1}-3}{2} + \frac{2}{2} \\ &= \frac{3^{d(T)+1}-1}{2}, \end{aligned}$$

as required.

Solution. Here is a proof using strong induction on size. (Previous solution to same problem was via structural induction.).

(a)

Solution. $P(n)$ is the statement: "If $T \in \mathcal{T}$ has size $|T| \leq n$, then $|T| \leq \frac{3^{d(T)+1}-1}{2}$."
n is any positive natural

(b)

Solution. The base case is $n = 1$. Let T_0 be the one node tree.
This is the only tree with size at most 1.
Since $d(T_0) = 0$, $|T| = 1 = \frac{3-1}{2} = \frac{3^{d(T_0)+1}-1}{2}$ so $P(1)$ is correct

(c) **Solution.**

Let $n > 1$. Assume that $P(i)$ is true for $i = n$.

Let T be any RTT with $|T| = n + 1$.

Thus,

- T is a root connected to the roots of 3 other RTTs, L, M and R.
- From the construction, $|T| = 1 + |L| + |M| + |R|$
  and $d(T) = 1 + \max(d(L), d(M), d(R))$.
- Since $|L|, |M|, |R|$ are all at most $n + 1 - 3 = n = 2$, we know that $P(|L|)$, $P(|M|)$, and $P(|M|)$ are all true by the IH,
- Thus,
$$|L| \leq \frac{3^{d(L)+1} - 1}{2}, \quad |M| \leq \frac{3^{d(M)+1} - 1}{2} \quad \text{and} \quad |R| \leq \frac{3^{d(R)+1} - 1}{2}$$
- Since $d(L) + 1 \leq d(T)$, $d(M) + 1 \leq d(T)$, and $d(R) + 1 \leq d(T)$, this implies
$$|L| \leq \frac{3^{d(T)} - 1}{2}, \quad |M| \leq \frac{3^{d(T)} - 1}{2} \quad \text{and} \quad |R| \leq \frac{3^{d(T)} - 1}{2}.$$

Combining the pieces above gives

$$\begin{aligned} |T| &= 1 + |L| + |M| + |R| \\ &\leq 1 + 3\left(\frac{3^{d(T)} - 1}{2}\right) \\ &= \frac{3^{d(T)+1} - 3}{2} + \frac{2}{2} \\ &= \frac{3^{d(T)+1} - 1}{2}, \end{aligned}$$

We have just seen that $|T| \leq \frac{3^{d(T)+1}-1}{2}$ for every $T \in \mathcal{T}$ with $|T| = n + 1$.

$P(n)$ already told us that $|T| \leq \frac{3^{d(T)+1}-1}{2}$ for every $T \in \mathcal{T}$ with $|T| \leq n$.

So $P(n + 1)$ is true.

Solution. Here is a proof using strong induction on depth. (Previous solutions to same problem were via structural induction and induction on size).

(a)

Solution. $P(n)$ is the statement: "If $T \in \mathcal{T}$ has depth $d(T) \leq d$, then $|T| \leq \frac{3^{d+1}-1}{2}$."
d is any natural.

(b)

Solution. The base case is $d = 0$. Let T_0 be the one node tree.
This is the only tree with depth 0.
Since $d(T_0) = 0$, $|T| = 1 = \frac{3-1}{2} = \frac{3^{d(T_0)+1}-1}{2}$ so $P(1)$ is correct

(c) **Solution.**

Let $d > 1$. Assume that $P(d)$ is true.

Let T be any RTT with $d(T) \leq d + 1$. Then

- T is a root connected to the roots of 3 other RTTs, L, M and R.
- From the construction, $|T| = 1 + |L| + |M| + |R|$
  and $1 + \max(d(L), d(M), d(R)) = d(T) \leq d + 1$
- Thus $d(T_L), d(T_M), d(T_r) \leq d$
- Thus, by the IH,

$$|L| \leq \frac{3^{d+1}-1}{2}, \quad |M| \leq \frac{3^{d+1}-1}{2} \quad \text{and} \quad |R| \leq \frac{3^{d+1}-1}{2}$$

Combining the pieces above gives

$$\begin{aligned} |T| &= 1 + |L| + |M| + |R| \\ &\leq 1 + 3\left(\frac{3^{d+1}-1}{2}\right) \\ &= \frac{3^{d+2}-3}{2} + \frac{2}{2} \\ &= \frac{3^{(d+1)+1}-1}{2}, \end{aligned}$$

We have just seen that $|T| \leq \frac{3^{(d+1)-1}-1}{2}$ for every $T \in \mathcal{T}$ with $d(T) \leq d + 1$.

So $P(n + 1)$ is true.

Marking Notes:

1. "Using Illegal Information" or "Expanding RTT" 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 RTT by 3^d. Again, while this is a correct statement, it would first have to be proven, *by induction* before it could be used.

Yet another similar error was one that discussed the *worst-case* scenario for adding nodes. Again while this might have been a correct statement the fact that it is worst-case would have to be proven by induction before it could be used.

2. Bottom-up Error.

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

3. $P(T+1)$ 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.

4. $|T| = |T_L| + |T_M| + |T_R|$ 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 not there, then the best inequality the resulting induction could derive would be

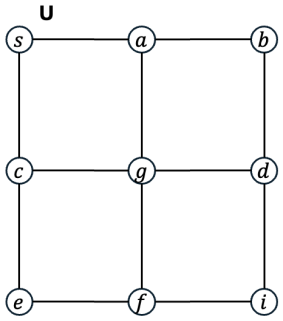
$$|T| \leq 3^{d(T)}$$

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

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 a labeled directed graph G. All represent the same nine nodes, and each has the same twelve edges, but they have the edge types appropriate to their type of graph.

- (A, 10) **Undirected Breadth-First Search:**
  Carry out a **BFS search** for the **undirected** graph U, starting with node s and with **no goal node**. List the order of each event in which a node either is placed on the open list or comes off the open list. Describe nodes 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.



With list items denoted as specified above, here is the combined list of events on the queue. We were flexible about your notation.

$(s, -)$ **starts on the list:**
$(s, -)$ **off;**
(a, s) **and** (c, s) **on;**
(a, s) **off;**
(b, a) **and** (g, a) **on;**
(c, s) **off;**
(e, c) **and** (g, c) **on;**
(b, a) **off;**
(d, b) **on;**
(g, a) **off;**
(d, g) **and** (f, g) **on;**
(g, c) **discarded;**
(e, c) **off;**
(f, e) **on;**
(d, b) **off;**
(f, d) **on;**
(d, g) **discarded;**
(f, g) **off;**
(e, f) **and** (i, f) **on;**

(f, e) **discarded;**
(i, d) **off;**
(i, f) **discarded;**

Also, draw the BFS tree from this search, indicating the tree edges and the non-tree edges, starting with node s and with no goal node.

All vertical edges (in the diagram above) are directed downward and are tree edges. Edges (s, a) **and** (a, b) **are directed to the right and are also tree edges. The other four edges are directed to the right and are non-tree edges.**

The mean was 8.68/10, with 55% getting full credit. The most common mistakes were missing non-edges in the tree, and getting node f as a child of e rather than of g. We noted that you had a choice of whether to put a new list item on for a node that was in the open list, but not the closed list. This choice in general did not affect the BFA tree.

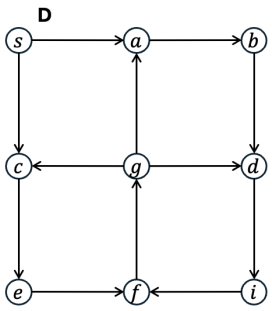
(B, 10) **Directed Depth-First Search:**

Carry out a **DFS search** for the **directed** graph D, starting with node s and **with no goal node**. List the order of each event in which a node either is placed on the open list or comes off the open list. Describe nodes 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.

Here is the combined list of events, using the same notation, using the procedure from the generic search algorithm:

$(s, -)$ **begins on the list.**
$(s, -)$ **comes off.**
(a, s) **and** (c, s) **go on.**
(a, s) **comes off.**
(b, a) **and** (g, a) **go on.**
(b, a) **comes off.**
(d, b) **goes on.**
(d, b) **comes off.**
(i, d) **goes on.**
(i, d) **comes off.**
(f, i) **goes on.**
(f, i) **comes off.**
(g, f) **goes on.**
(g, f) **comes off.**
(a, g) **and** (d, g) **are recognized as back edges.**
(c, g) **goes on.**
(c, g) **comes off.**
(e, c) **goes on.**
(e, c) **comes off.**
(f, e) **is recognized as a back edge.**
(c, s) **comes off and is recognized as a forward edge.**



Also, draw the DFS tree for this search, indicating the tree and non-tree edges. For each non-edge, **identify** it as a back, cross, or forward edge.

The eight tree edges form a linear path containing all nine nodes in the order s-a-b-d-i-f-g-c-e. There are back edges from g to a, from g to d, and from e to f. There is a forward edge from s to c. There are no cross edges.

We also allowed a procedure where list items for nodes on the open list (but not on the closed list) are not put on the stack. This changes the combined list as follows:

$(s, -)$ **begins on the list.**
$(s, -)$ **comes off.**
(a, s) **and** (c, s) **go on.**
(a, s) **comes off.**
(b, a) **and** (g, a) **go on.**
(b, a) **comes off.**
(d, b) **goes on.**
(d, b) **comes off.**
(i, d) **goes on.**
(i, d) **comes off.**
(f, i) **goes on.**
(f, i) **comes off.**
(g, f) **goes on.**
(g, f) **comes off.**
(a, g) **and** (d, g) **are recognized as back edges.**
(c, g) **is not put on the list because** c **is on the open list.**
(c, s) **comes off.**
(e, c) **goes on.**
(e, c) **comes off.**
(f, e) **is recognized as a cross edge.**
The stack is empty and the search terminates.

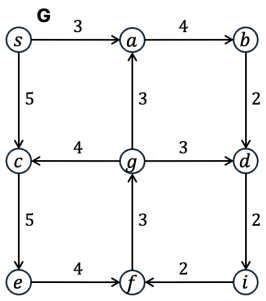
The new tree has one linear branch with nodes s-a-b-d-i-f-g and another branch with nodes s-c-e. There are still back edges (g, a) **and** (g, d)**, but now** (g, c) **and** (e, f) **are cross edges. There are no forward edges.**

The mean was 8.09, with 34% getting full credit. Either tree was ok, but I took off a point if your tree did not follow the sequence of events that you reported. There were lots of minor mistakes with the non-tree edges, such as forgetting one of them. If your tree was correct, I usually did not look too closely at your sequence of events, so there are varying degrees of detail in full-credit answers.

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

Conduct a uniform-cost search of the labeled directed graph G, with s as the start node and g as the goal node. Indicate when each item enters and leaves the priority queue. In each case when the first item for a node leaves the priority queue, report the priority value for for that node.

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



Here is the combined list of events, using the suggested notation:

$(s, 0, 1)$ **starts on the list, and comes off.**
$(a, 3, s)$ **and** $(c, 5, s)$ **on.**
$(a, 3, s)$ **off.**
$(b, 7, a)$ **on.**
$(c, 5, s)$ **off.**
$(e, 10, c)$ **on.**
$(b, 7, a)$ **off.**
$(d, 9, b)$ **on.**
$(d, 9, b)$ **off.**
$(i, 11, d)$ **on.**
$(e, 10, c)$ **off.**
$(f, 14, c)$ **on.**
$(i, 11, d)$ **off.**
$(f, 13, i)$ **on.**
$(f, 13, i)$ **off.**
$(g, 16, f)$ **on.**
$(f, 14, i)$ **discarded, as** f **is on the closed list.**
$(g, 16, f)$ **off.**
The search ends with g **found using the path s-a-b-d-i-f-g of length 16.**

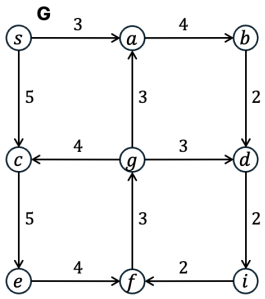
The mean was 8.48, with 58% getting full credit. There was a three-point penalty for not getting the answer of distance 16 *— I tried to determine where you went wrong. If you were doing a UCS made a mistake, you got most of the remaining points, but if you weren't, I was more harsh. (For example, if you found the right path without examining node e, you were not doing UCS.) I was more casual about the endgame, though I took off a point if you took* $(f, 14, i)$ *off the list, or if you continued after taking* $(g, 16, f)$*. I should probably have looked harder to penalize stopping the search when you* first *found* $(g, 16, f)$*, but this often looked like the correct sequence.*

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

We will now ask you to conduct an A^* search for the directed graph G, with s as the start node and g as the goal node, using the following heuristic function h. For any node x, $h(x)$ will be the distance from x to g *ignoring the direction of the edges*. We could have you calculate that by a UCS of the undirected version of the graph, but we will just give you the answers:

$$\begin{array}{llllllllll} h(s) & = & 6, & h(a) & = & 3, & h(b) & = & 5, & h(c) & = & 4, & h(g) & = & 0 \\ h(d) & = & 3, & h(e) & = & 7, & h(f) & = & 3, & h(i) & = & 5 \end{array}$$

Conduct an A^* search of the directed graph D to determine the shortest distance from s to g, using the heuristic function h above. Indicate which items enter and leave the priority queue. Whenever the first item for a node comes off of the priority, report the distance from the start node to that node. When two or more nodes need to come off the list, they should follow the rules of an A^* search, and you may break ties as you like.



Here is the combined sequence of events, using the suggested notation:

$(s, 6, 0, -)$ **starts on the PQ and comes off.**
$(a, 6, 3, s)$ **and** $(c, 9, 5, s)$ **go on.**
$(a, 6, 3, s)$ **comes off.**
$(b, 12, 7, a)$ **goes on.**
$(c, 9, 5, s)$ **comes off.**
$(e, 17, 10, c)$ **goes on.**
$(b, 12, 7, a)$ **comes off.**
$(d, 12, 9, b)$ **goes on and comes off.**
$(i, 16, 11, d)$ **goes on and comes off.**
$(f, 16, 13, i)$ **goes on and comes off.**
$(g, 16, 16, f)$ **goes on and comes off, and the search ends as** g **is the goal node.**
We have found the path s-a-b-d-i-f-g of length 16. Compared with the UCS from Q3C, we have saved some time because we didn't have to explore node e**.**

The mean was 8.05.10, with 63% full-credit answers. I took off three points for getting the wrong answer, or no answer, plus whatever else was wrong. I needed to see that you were actually carrying out Aj, which meant that node $(e, 17, 10, c$ *went on the open list and stayed there for the whole search. The comment "runaway numbers" refers to a particular problem where you keep adding multiple h values to a single priority number, getting values that are too large.*

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	T	T	T	F	F	T	T	T	F

- (a) Let $A \subseteq \{0,1\}^*$ be a language defined by the two rules $\lambda \in A$ and $\forall w : (w \in A) \rightarrow [(awa \in A) \wedge (bwb \in A)]$, such that the only strings in A are those given by those rules. Then there does not exist any string in A with odd length.

  TRUE (65% correct). By induction on the rules, every string in A has even length.

- (b) Let $P(n)$ be a predicate on the naturals. If $P(0)$ is true, and $\forall n : P(n) \rightarrow (P(3n) \wedge P(3n+1) \wedge P(3n+2))$ is true, then $P(n)$ is true for all naturals n.

  TRUE (70% correct). We can prove this by strong induction, as any case $P(n+1)$ follows from some case $P(m)$, where $m = (n+1)/3$ using integer division.

- (c) Let T be a boolean expression tree, where the only operators used are $\neg$ and $\oplus$. If we change the value of one of its leaves from true to false, or vice versa, then the value of T also changes from true to false, or vice versa.

  TRUE (78% correct). Because $\neg x$ has the opposite value from $\neg(\neg x)$, and $x \oplus y$ has the opposite value from either $x \oplus (\neg y)$ or $(\neg x) \oplus y$, we can prove this on all trees following this definition.

- (d) Let u and v be two distinct nodes in a directed graph G such that there exists a directed path from u to v. If all directed edges have the same cost, then we can use BFS starting at u to find the path from u to v with the smallest cost.

  TRUE (74% correct). If there is a path at all, the BFS path will find a path with the minimum number of edges in it, and this path will have the smallest cost of any path.

- (e) If $h(x)$ is an admissible heuristic function for an A^* search of an undirected labeled graph, then $h(x)$ must be at least as large as the shortest-path distance from x to the goal node.

  FALSE (68% correct). It may be smaller than that distance, but it must be no *larger*.

- (f) Let T be any tree, and consider making a rooted tree from it by choosing one of its nodes as the root. Then if v has strictly more neighbors than any other node, choosing v as the root gives a rooted tree with the smallest possible depth.

  FALSE (73% correct). One counterexample has edges (a,b), (b,c), (b,d), (d,e), and (e,f). Then b, with three neighbors, would be chosen and has depth 3. But choosing d would give a depth of 2.

- (g) Consider the set Σ^* of all strings over the alphabet $\Sigma = \{a\}$. Then the concatenation operations for such strings is both commutative and associative.

  TRUE (80% correct). We proved that string concatenation is associative over any alphabet. String concatenation is not commutative over an alphabet with more than one letter but over this alphabet, the only strings are of the for a^i for some natural i. And in this case $a^i \circ a^j$ and $a^j \circ a^i$ are the same string a^{i+j}.

- (h) Consider undirected graphs with exactly four nodes. Any such graph with exactly four edges is connected, but there exist such graphs with four nodes and three edges that are not connected.

  TRUE (70% correct). If a node is isolated, there can be at most three remaining edges, and if there are two components of two nodes each, there are only two edges. A graph with a triangle and an isolated node has exactly three edges, and is not connected.

- (i) Consider a game tree (as defined in Lecture 27),where there is exactly one leaf marked "B" for Black victory, all the other leaves are marked "W" for White victory, the root is not a leaf, every internal node has exactly two children, and White moves first. Then White has a winning strategy for this game.

  TRUE (78% correct). White has two options on her first move, and only one could lead to a Black victory. If she takes the other move, she is sure to win.

- (j) Consider the boolean expression with infix representation $(a \wedge b) \vee (c \wedge d)$. Then the prefix and postfix representations of this expression are reversals of one another.

  FALSE (84% correct). The prefix is $\vee \wedge ab \wedge cd$ and the postfix $ab \wedge cd \wedge \vee$. They are not reversals of one another, though they are close.

The overall mean was 14.8/20*, with 14% perfect 20/20 scores.*