NAME: ____________________________

SPIRE ID: ____________________________

COMPSCI 250
Introduction to Computation
Second Midterm Fall 2024

D. A. M. Barrington and M. Golin 7 November 2024

DIRECTIONS:

- Answer the problems on the exam pages.
- There are five problems on pages 2-12, some with multiple parts, for 100 total points plus 10 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 have a note on the front stating that you used the back page.
- 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	/10
2	/20+10
3	/10
4	/40
5	/20
Total	/100+10

Family Name: ____________________

Question 1 (10): (Induction 1)

For positive naturals n, let $S(n)$ be the sum $\frac{1}{2\cdot 3} + \frac{1}{3\cdot 4} + \frac{1}{4\cdot 5} + \cdots + \frac{1}{(n+1)(n+2)}$.

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

Examples: $S(1) = \frac{1}{2\cdot 3} = \frac{1}{6}$, $S(2) = \frac{1}{2\cdot 3} + \frac{1}{3\cdot 4} = \frac{3}{12} = \frac{1}{4}$, $S(3) = \frac{1}{2\cdot 3} + \frac{1}{3\cdot 4} + \frac{1}{4\cdot 5} = \frac{18}{60} = \frac{3}{10}$, etc..

Prove *by induction* that for all positive naturals n, $S(n) = \frac{n}{2(n+2)}$.

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 $P(x)$.

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

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

Family Name: ____________________

Question 2 (20+10): (Induction 2) A **rooted binary-ternary tree (RBTT)** 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 two other RBTT's or

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

Furthermore,

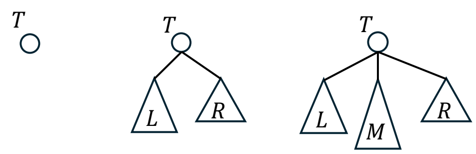
R3: The only RBTT's are those made by the first three rules.

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

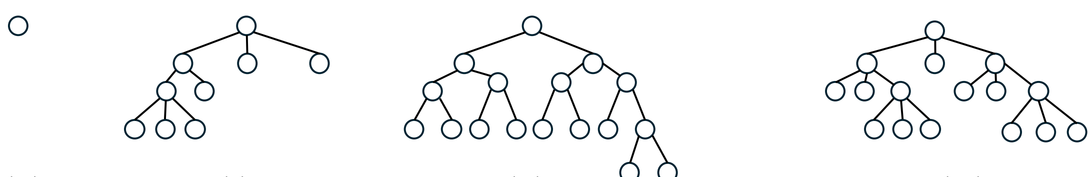
$$\begin{aligned} |T| &= \text{the number of nodes in } T, \\ L(T) &= \text{the number of leaves in } T, \\ I(T) &= \text{the number of internal nodes in } T, \\ d(T) &= \text{the depth of } T. \end{aligned}$$

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 RBTT's and their associated values.



(i) (ii) (ii)



(iv)	(v)	(vi)	(vii)
$L(T) = 1$	$L(T) = 6$	$L(T) = 9$	$L(T) = 11$
$I(T) = 0$	$I(T) = 3$	$I(T) = 8$	$I(T) = 5$
$d(T) = 0$	$d(T) = 3$	$d(T) = 4$	$d(T) = 3$
$\|T\| = 1$	$\|T\| = 9$	$\|T\| = 17$	$\|T\| = 16$

Parts (A)-(C) of this problem are on the following pages ((C) is for extra credit). When writing the solutions to parts (A)-(C), 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) Prove by induction that every $T \in \mathcal{T}$ satisfies

$$d(T) \leq L(T) - 1.$$

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) Prove by induction that every $T \in \mathcal{T}$ satisfies

$$L(T) \leq 3^{d(T)}.$$

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: ____________________

(C) Extra Credit Problem. Prove by induction that every $T \in \mathcal{T}$ satisfies

$$I(T) + 1 \leq L(T) \leq 2I(T) + 1.$$

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: ____________________

Question 3 (10) (Expression Trees)

In this problem we deal with **boolean expression trees** as in Discussion #7, where we have a unary operator $\neg$, two binary operators $\wedge$ and $\vee$, and constants 0 and 1. Recall the definitions of the **infix**, **prefix**, and **postfix** strings for any expression.

- (a, 2) Write the prefix string for the boolean expression with infix string (no justification needed):
  $\neg((1 \vee (1 \wedge (1 \vee 0))) \wedge (\neg(1 \wedge 0) \wedge (1 \wedge \neg 1)))$

  The parentheses are only provided to make the expression unambiguous. They should *not* appear in the prefix string you write out or in the postfix string in part (b).

- (b, 2) Write the postfix string for the same boolean expression (no justification needed):

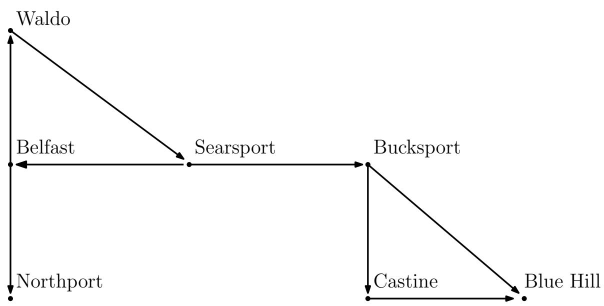
- (c, 6) Argue convincingly that the prefix string of *any* boolean expression ends with a constant and (unless it has only one letter) begins with an operator. [Hint: Consider the recursive definition of the prefix string.]

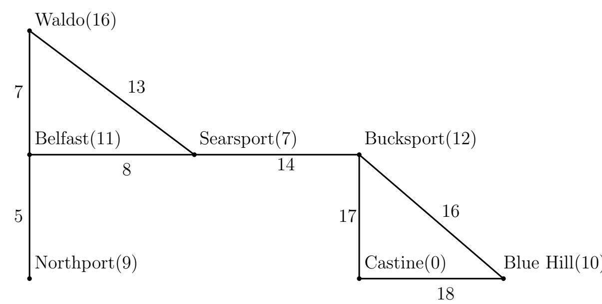
Family Name: ____________________

Question 4 (40) (Search Algorithms)

This problem deals with the road network connecting seven cities in Maine. Our first diagram D, the one on the left, is a *directed* graph with seven nodes and eight directed edges. We will also use the *undirected* graph U obtained from D by making all the edges two-way.

The second diagram L, the one on the right, is a *labeled undirected* graph on the same cities where the label on each edge is the driving distance (in miles) from one city to the other. The *node* labels, in parentheses to the right of the city name, represent the crow-flies distance (in miles) between that city and Castine.





(A) Undirected Breadth-First Search

Carry out a breadth-first search of the undirected graph U, with Belfast as the start node and with no goal node. If two or more nodes go on to the queue at the same time, they come off in alphabetical order with respect to the city name. Indicate the order in which the nodes are placed on the queue.

Draw a BFS tree, indicating which are tree and which are non-tree edges. Include all the edges of the graph in your drawing.

Family Name: ____________________

(B) Directed Depth-First Search

Carry out a depth-first search of the directed graph D, with Belfast as the start node and with no goal node. If two or more nodes go on to the stack at the same time, they come off in alphabetical order with respect to the city name. Indicate the order in which the nodes are placed on the stack.

Draw a DFS tree, and indicate the type (tree, back, cross, or forward) for each edge. Include all the edges of the graph in your drawing.

(C) Uniform-Cost Search

Carry out a uniform-cost search for the labeled graph L, with Belfast as the start node and Castine the goal node. Indicate the order in which the nodes are placed on the priority queue. Use a closed list, so that an entry for a node should not be placed onto the priority queue if an entry for that node has already been *removed* from the priority queue. But note that there may be multiple nodes on the priority queue for the same city.

(D) A^* Search

Carry out an A^* search for the labeled graphs, with Belfast as the start node and Castine as the goal node, using the heuristic value $h(x)$ for each node x given by the node labels. Indicate the order in which the nodes are placed on the priority queue, with the priority for each node. Use a closed list as described in part (c). But note that there may be multiple nodes on the priority queue for the same city.

Question 5 (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. Be sure that your "T" and "F" characters are consistent and distinct.

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

- (a) Let k be any positive natural, and let X be the set $\{0, 1, \ldots, k-1\}$. Define the operations of successor, addition, and multiplication using arithmetic modulo k. Then the resulting system of arithmetic satisfies the Peano Axioms for naturals.

- (b) Let $P(w)$ be a predicate on binary strings. If we prove $P(\lambda)$ and $\forall w : P(w) \rightarrow (P(w0) \wedge P(w10) \wedge P(w11))$, we may conclude $\forall w : P(w)$.

- (c) Let $P(n)$ be a predicate on the naturals. If we prove $\forall n : P(n) \rightarrow P(n+3)$, we can complete the proof with three base cases $P(0)$, $P(1)$, and $P(2)$.

- (d) Let D be a directed graph with three nodes and no self-loops. If D has at least five directed edges, it is strongly connected, but if it has only four edges it may fail to be strongly connected.

- (e) Let G be a connected undirected graph that has exactly one simple cycle. Then if we remove any one of the edges in G, the resulting graph will be a tree.

- (f) Consider tiling a $2 \times n$ rectangle with L-shaped tiles as in Lecture 20. Then it is not the case that this can be done if and only if n is divisible by 3.

- (g) Consider breadth-first and depth-first searches of a directed acyclic graph G, where there is no closed list or other means to detect visited nodes. Assume that there is some path from the start to the goal node. Then the BFS is guaranteed to find the goal node, but the DFS is not.

- (h) In a breadth-first search tree for a directed graph, every non-tree edge must go from a node to one of its ancestors in the tree.

- (i) Let H be a labeled directed graph where every directed edge has cost 3. Let x and y be two nodes of H such that there is at least one directed path from x to y. Then a breadth-first search from x can be used to find the minimum-cost path from x to y.

- (j) Let G be a two-player deterministic game defined by a finite game tree, where each leaf is labeled either as a White win, a draw, or a Black win. Then either White has a winning strategy for G or Black has a strategy that guarantees White will not win.