MAT385 Final, Spring 2022

Name:

Directions:

- Each problem is worth 10 points. **You must skip one of the "old" problems (1-6), and one of the new problems, 7-13.** Write "SKIP" clearly on each problem you skip.
- Show your work! Answers without justification will likely result in few points. **Good luck, and have a wonderful summer.**

Problem 1: The word "lexicography" is a long one without repeated letters.

a. (4 pts) Create the binary search tree by entering the 12 letters of lexicography in word order.

b. (2 pts) Does this tree meet minimum depth among binary search trees of 12 letters? Explain.

c. (4 pts) Write the following traversals (Which new word is easier to pronounce?:):

- pre-order:
- post-order:

Problem 2: Prove the formula for a geometric series ($r \neq 1$): $\sum_{i=0}^{n} r^i = \frac{1 - r^{n+1}}{1 - r}$

Problem 3: Logic. Use propositional or predicate logic to prove the following, using the statement/predicate letters (F)inish, (C)andy, (E)xercise, (G)ain:

a. (5 pts) If I finish grading, I eat a candy bar; if I eat a candy bar, I exercise or gain weight; I did not gain weight nor did I exercise. Therefore I did not finish grading.

b. (5 pts) Anyone who finishes grading eats a candy bar; if anyone eats a candy bar, they exercise or gain weight; Andy did not gain weight, nor did he exercise. Therefore someone did not finish grading.

Problem 4: In this problem we consider full binary trees.

a. (4 pts) Write a linear recurrence relation for the number of leaves $L(n)$ of a full binary tree of depth n.

b. (2 pts) Give the simplest closed form solution for $L(n)$.

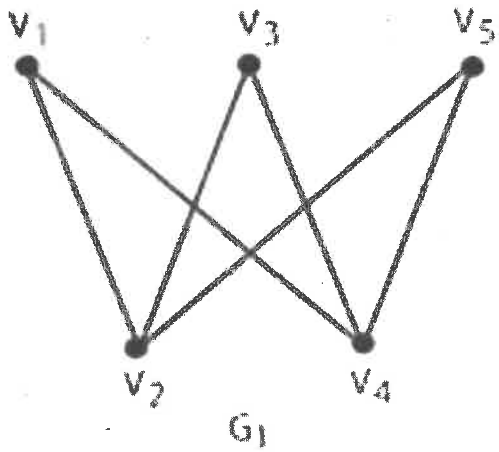
c. (4 pts) Give a recursive explanation for why the number of nodes in a full binary tree of depth n is the sum of all the leaves of trees of depths 0 to n

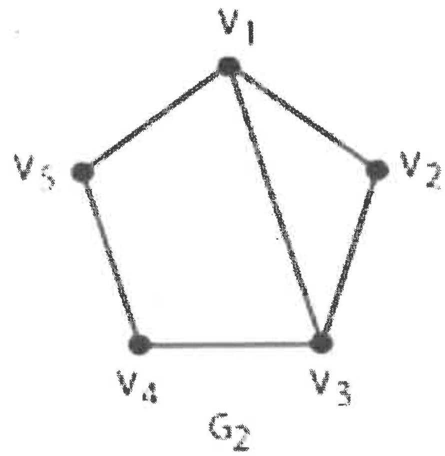
$$\sum_{i=0}^{n} L(i)$$

Then determine the number of nodes in a full binary tree by computing the sum, using your answer from part b.

Problem 5: Isomorphisms and bijections (one-to-one and onto mappings).

a. (4 pts) The two graphs shown are not isomorphic: identify a characteristic of the graphs which demonstrates that they are not isomorphic.





b. (4 pts) If you add enough edges to each, you'll get two complete simple graphs, which are isomorphic. What is the **minimal number** of edges you must add to create isomorphic graphs? Justify!

c. (2 pts) Is there is a bijection between the set of natural numbers and the set of rational numbers? If so, what does this prove? If not, what does this prove?

Problem 6:

a. (4 pts) Write the power set of the set of letters $\{a, b, c, d\}$.

b. (4 pts) Relate the elements of the power set to a row in Pascal's triangle:

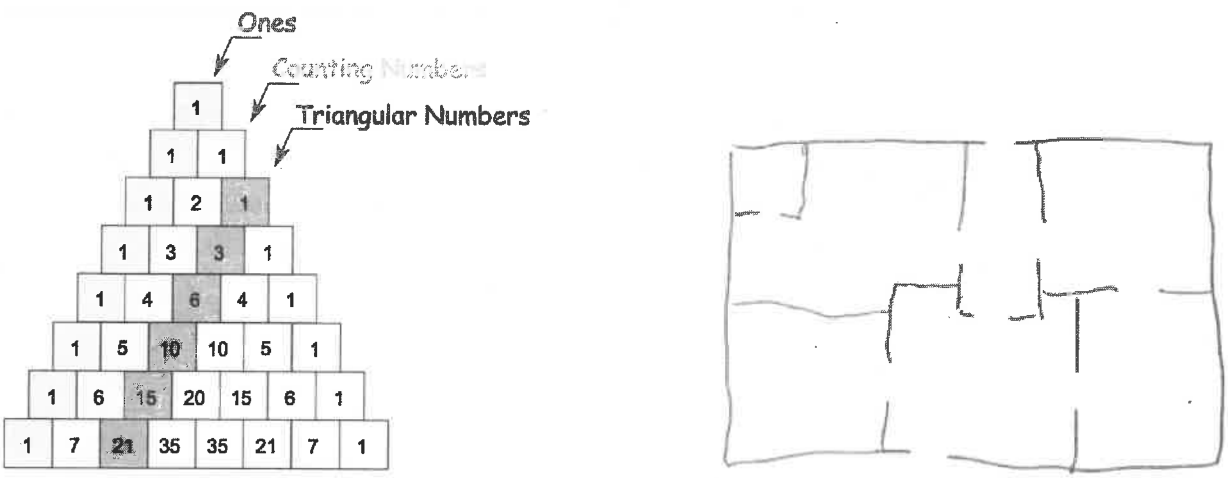


Figure 1: At left, Pascal's triangle; at right, floor plan for Problem 7.

c. (2 pts) Explain why the rows of Pascal's triangle sum to powers of 2.

Problem 7:

a. Consider the house floor plan in the figure above. Treating the rooms as nodes, and the doors as edges, draw a corresponding graph (superimposed on the floor plan).

b. (5 pts) If this graph is planar, demonstrate Euler's formula; if not, explain why not.

c. (5 pts) Is it possible to pass through each door exactly once before re-using any door? Explain.

Problem 8: Here is the Adjacency matrix for a graph G:

	a	b	c	d
a	0	5	1	∞
b	5	0	2	1
c	1	2	0	∞
d	∞	1	∞	0

a. (2 pts) Draw a corresponding graph of G.

b. (4 pts) Use Dijkstra's algorithm to compute the shortest distance and path from node a to node d

	a	b	c	d
d				
s				

	a	b	c	d
d				
s				

	a	b	c	d
d				
s				

c. (2 pts) How is the path related to a minimal spanning tree for G?

d. (2 pts) Under what conditions will the shortest path between two nodes produce a minimal spanning tree for a weighted graph?

Problem 9: Graph traversals on complete graphs K_n are kind of boring.

a. (4 pts) As an example, draw K_4, with nodes $\{a, b, c, d\}$, and its adjacency matrix. In general, what does the adjacency matrix of a complete graph look like?

b. (2 pts) If we do a depth-first traversal of the graph starting from node a, what do we get? Anything different about a **breadth-first** traversal, starting from node a?

c. (2 pts) What results if we do a depth-first traversal of the graph starting from any other node?

d. (2 pts) Describe a recursive method for doing depth-first traversal of complete graphs K_n with nodes labelled $\{x_1, x_2, \ldots, x_n\}$.

Problem 10: Prove the following two properties of a Boolean Algebra, giving reasons for each step:

a. $(x + y) \cdot (x' + y) = y$

Write the dual property: $\boxed{}$

b. $x \cdot y + x' = y + x' \cdot y'$

Write the dual property: $\boxed{}$

Problem 11: Consider the canonical sum of products

$$f(x_1, x_2, x_3, x_4) = x_1x_2x_3x_4 + x_1x_2'x_3x_4 + x_1'x_2x_3x_4 + x_1x_2x_3'x_4 + x_1x_2'x_3x_4' + x_1'x_2x_3'x_4$$

a. (2 pts) Draw the corresponding truth function:

x_1	x_2	x_3	x_4	$f(x_1, x_2, x_3, x_4)$
1	1	1	1	
1	1	1	0	
1	1	0	1	
1	1	0	0	
1	0	1	1	
1	0	1	0	
1	0	0	1	
1	0	0	0	
0	1	1	1	
0	1	1	0	
0	1	0	1	
0	1	0	0	
0	0	1	1	
0	0	1	0	
0	0	0	1	
0	0	0	0	

b. (4 pts) Create the Karnaugh map for that sum

	x_1x_2	x_1x_2'	$x_1'x_2'$	$x_1'x_2$
x_3x_4				
x_3x_4'				
$x_3'x_4'$				
$x_3'x_4$				

and write the resulting minimal sum of products.

c. (4 pts) Minimize it some more: show that you need no more than 7 of the logic elements ($+$, $\cdot$, $'$). Then draw a logic network for the minimized sum of products.

Problem 12: Consider the (same) canonical sum of products

$$x_1x_2x_3x_4 + x_1x_2'x_3x_4 + x_1'x_2x_3x_4 + x_1x_2x_3'x_4 + x_1x_2'x_3x_4' + x_1'x_2x_3'x_4$$

a. (6 pts) Use Quine-McCluskey to find a minimal sum of products form for the Boolean expression. Here's the set up for the first round:

# of 1s	x_1	x_2	x_3	x_4
4	1	1	1	1
3	1	0	1	1
	0	1	1	1
	1	1	0	1
2	1	0	1	0
	0	1	0	1

# of 1s	x_1	x_2	x_3	x_4
3				
2				

# of 1s	x_1	x_2	x_3	x_4
2				

b. (4 pts) Now it's on to round two: complete the table, and give the minimal expression. (If you did the

	1111	1011	0111	1101	1010	0101

previous problem, you should get (essentially) the same answer!)

Problem 13: Create

a. (2 pts) a regular expression for all strings ending in 101.

b. (8 pts) a finite state machine that recognizes such strings.