MAT385 Exam 2 (Spring 2025): Chapters 4-7

Name:

Directions:

Show your work: answers without justification will likely result in few points. Your written work also allows me the option of giving you partial credit in the event of an incorrect final answer (but good reasoning). Indicate clearly your answers to each problem (e.g., put a box around them). **Good luck!**

Problem 1: (10 pts) Do two of the following three. Write "skip" on the one you're skipping.

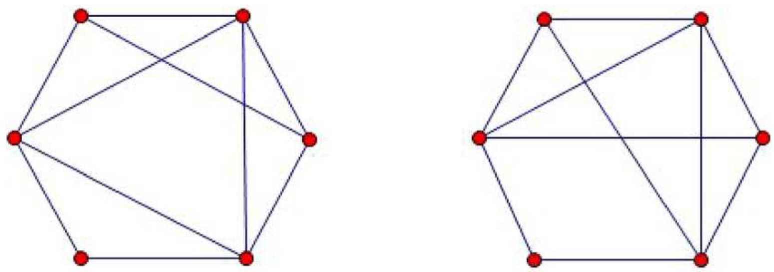
a. Write out all elements of the power set of the power set of the power set of the empty set. How many elements are there?

b. Prove that the set E of even natural numbers, $E \subset \mathbb{N}$, has the same cardinality as $\mathbb{N}$ itself.

c. Prove that if $\wp(A) \subseteq \wp(B)$, then $A \subseteq B$.

Problem 2: (10 pts) Do two of the following three. Write "skip" on the one you're skipping.

a. Determine whether these graphs are isomorphic. You must provide **justification**!



(a) **(b)**

b. Prove that $K_{2,3}$ is a planar graph.

c. The adjacency matrix for a directed graph is given below: draw the graph. It's planar, so verify that Euler's formula works for your graph.

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 2 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Problem 3: (10 pts) Do two of the following three. Write "skip" on the one you're skipping.

a. Draw a tree with preorder traversal a-b-d-e-h-f-c-g and postorder traversal d-h-e-f-b-g-c-a

b. Draw the binary tree corresponding to the left child-right child representation that follows (where node 1 is the root). Then write its inorder traversal.

Node	Left Child	Right Child
1	2	3
2	4	0
3	5	0
4	6	7
5	0	0
6	0	0
7	0	0

c. Two trees are *isomorphic* if there is a bijection $f : N_1 \to N_2$, where f maps the root of tree T_1 to the root of T_2, and where $f(y)$ is a child of $f(x)$ in T_2 when y is a child of x in T_1. Draw all the nonisomorphic trees with four nodes.

Problem 4: (10 pts) Given the data i, e, f, b, d, g.

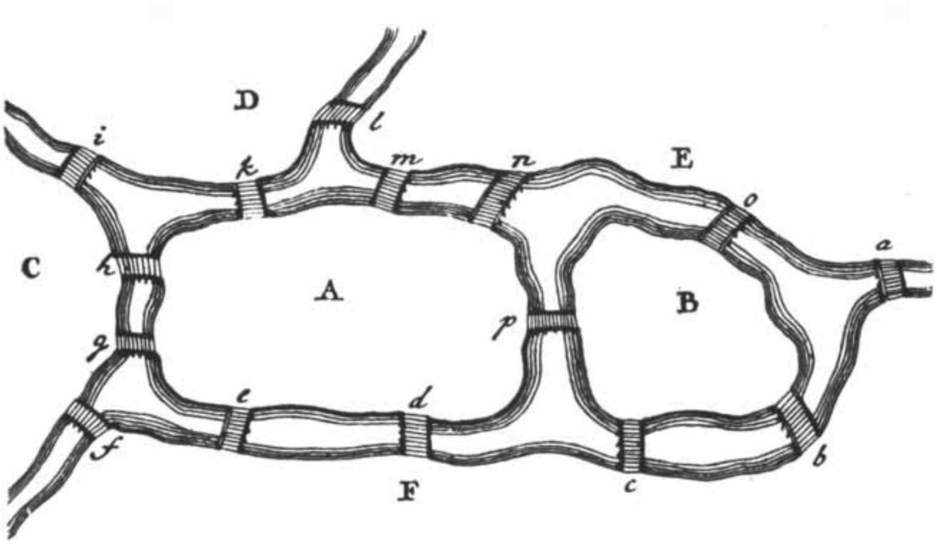
a. (3 pts) Construct the binary search tree with the data given in the order above. What is the depth of the tree?

b. (2 pts) What is the maximum number of comparisons done to search this tree for an item that is known to be in the list? Is this tree optimal for binary search on this data (explain)?

c. (5 pts) An inorder traversal of a binary search tree produces an ordered listing of the elements – i.e. sorts the data. So one idea for sorting is to fill a binary search tree with the elements in the order given, and then do an inorder traversal. What is the worst-case number of comparisons that one must do to fill a binary search tree with n elements? (You might look at the simplest cases, and generalize to n. You don't need to prove your conjecture, so long as it's right!) Is this optimal in its worst-case behavior for comparisons?

Problem 5: (10 pts)

a. (4 pts) Below is a figure that Euler studied for his work on the Bridges of Konigsberg problem. Does the associated graph have an Euler path? Explain.



b. (2 pts) Euler would have drawn fifteen arcs to create the graph corresponding to the figure above, with six nodes. If we draw it as a planar graph, without false intersections, how many regions will the plane be divided into?

c. (4 pts) Recall that that K_n denotes the simple, complete graph of order n.

i. For what values of n does an Euler path exist in K_n?

ii. For what values of n does a Hamiltonian circuit exist in K_n?

Problem 6: (10 pts) Find the shortest distance between the nodes a and e in the following graph by using one (and only one!) of our three algorithms (Dijkstra's, Bellman-Ford, or Floyd's). You must **demonstrate the algorithm** – graphically, as we did in class, or by using our author's methods.

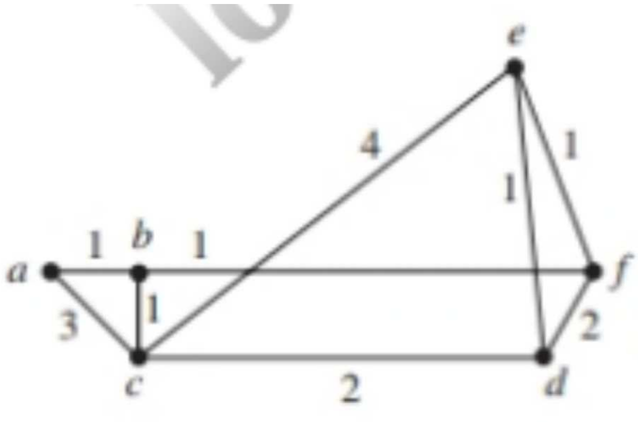


Figure 1: The graph

$$\begin{pmatrix} 0 & 1 & 3 & \infty & \infty & \infty \\ 1 & 0 & 1 & \infty & \infty & 1 \\ 3 & 1 & 0 & 2 & 4 & \infty \\ \infty & \infty & 2 & 0 & 1 & 2 \\ \infty & \infty & 4 & 1 & 0 & 1 \\ \infty & 1 & \infty & 2 & 1 & 0 \end{pmatrix}$$

Figure 2: The adjacency matrix