Exam 3 CS 211 Spring 2018

Name: ______________________________

Calculations: Students may use a scientific calculator on the exam, but not a graphing calculator. Exam work must be solo-work.
The formula is more important than the computation.

Readability: Please write in a clear and linear manner. **Using a pencil is best**. If I am not able to interpret your answer, you might lose out on points.

Exam format: This exam covers topics from Chapter 6.

Grading: Each question can receive between 0 and 4 points, and each question has a weight associated with it. The point value is used to compute the score for a question. For example, if a question is worth a weight of 5% and the student receives 3 points, then that question will count for 3.75% out of the full 5%.

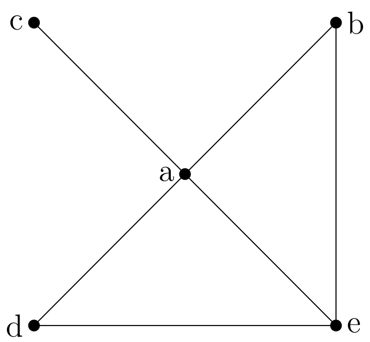
0	1	2	3	4
Nothing written	Attempted, but incorrect	Partially correct; multiple errors	Mostly correct, one or two errors	Perfect; correct answer & notation

#	Name	Weight	Points (out of 4)
1	Basic terms (4)	8%	
2	Terminology (8)	8%	
3	Cycles	10%	
4	Eulerian	10%	
5	Spanning Tree	14%	
6	Minimal Spanning Tree	16%	
7	Isomorphism (3)	9%	
8	Planarity	10%	
9	Adjacency Matrix	15%	
ex	Traversal	1%	

Exam

Chapter 7.1: Graph Theory

abc% Question 1: Basic Terms ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4



a. Identify all vertices (nodes):

b. Identify all edges:

c. Identify the degrees:

$deg(a) =$ $deg(b) =$

$deg(c) =$ $deg(d) =$

$deg(e) =$

d. What is the maximum degree?

abc% Question 2: Terminology ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4

Match the terms and the definitions.

a: ______ b: ______ c: ______ d: ______

Terms:

a. Loop

b. Parallel edge

c. Trail

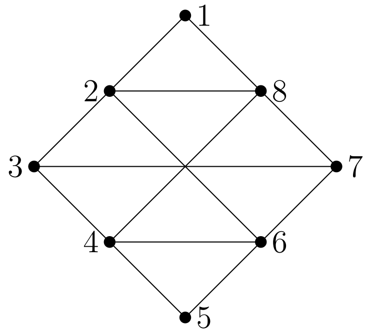
d. Connected graph

Definitions:

1. A walk with no repeated edges
2. A graph where there is a walk between any pair distinct nodes.
3. Two edges that have the same two endpoints
4. An edge that begins and ends at the same node

abc% Question 3: Cycles ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4

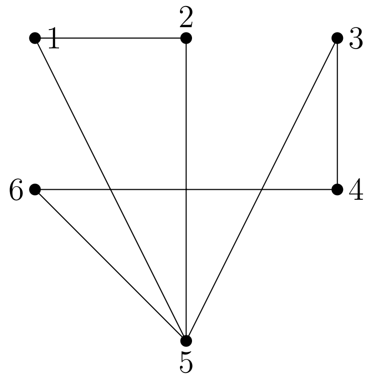
For the following graph:



Find a circuit of length 4 starting and ending at 8. List out all the nodes in the circuit, in order.

abc% Question 4: Connected, Eulerian □ 0 □ 1 □ 2 □ 3 □ 4

Identify a Eulerian [1] trail or circuit for the following graph. List out all the nodes in the trail/circuit.

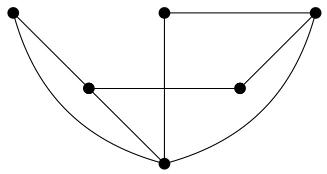


[1] A Eulerian trail/circuit is a trail or circuit is one where every edge is traversed

Chapter 7.2: Proofs About Graphs and Trees

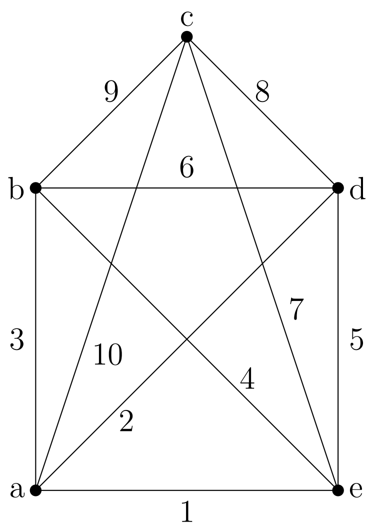
abc% Question 5: Spanning trees □ 0 □ 1 □ 2 □ 3 □ 4

Build out a spanning tree for the following graph.



abc% Question 6: Minimal spanning trees ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4

Starting at vertex **a**, find a minimal spanning tree for the following graph and specify the **total weight**. Label all vertices and edges in your minimal spanning tree.

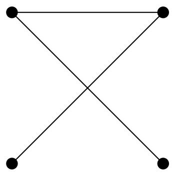


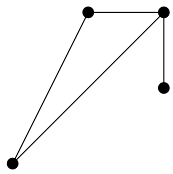
Chapter 7.3: Isomorphism and Planarity

abc% Question 7: Isomorphism (7.3) □ 0 □ 1 □ 2 □ 3 □ 4

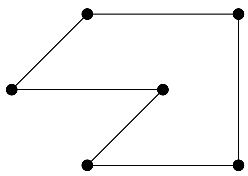
Determine if the following graph pairs are isomorphic. [2] You may use the graph maker at `https://rachels-courses.github.io/Visualizations/` for this question.

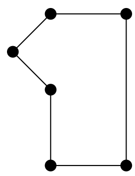
a.



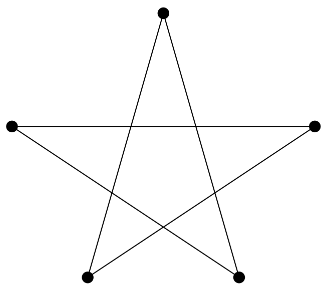


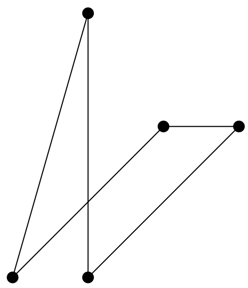
b.





c.

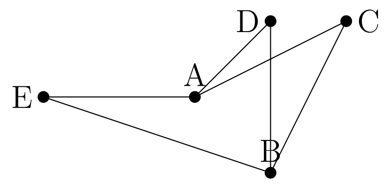




[2] Simple graphs G and H are isomorphic if there is a one-to-one and onto function f from the nodes of G to the nodes of H such that $\{v, w\}$ is an edge of G if and only if $\{f(v), f(w)\}$ is an edge of H.

abc% Question 8: Planarity (7.3) □ 0 □ 1 □ 2 □ 3 □ 4

Redraw the following graph to be Planar, [3] make sure to label the vertices. You may use the graph maker at `https://rachels-courses.github.io/Visualizations/` for this question.

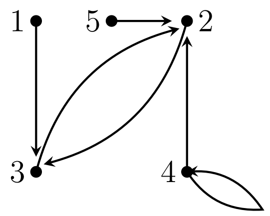


[3] A simple, connected graph is called planar if there is a way to draw it on a plane so that no edges cross

Chapter 7.4: Connections to Matrices and Relations

abc% Question 9: Adjacency matrix (7.4) ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4

Build the adjacency matrix for the following directed graph.



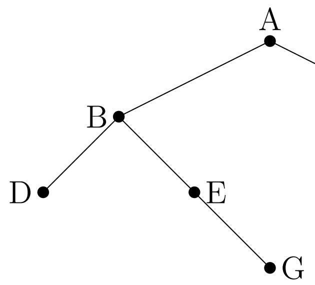
Columns

Rows	1	2	3	4	5
1					
2					
3					
4					
5					

Extra Credit

abc% Question 10: Extra credit ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4

Write out the pre-order traversal walk of this binary tree.



Cheat sheet

Spanning Tree Algorithm

- Begin with a simple, connected graph G_0.
- For each $i \geq 1$, as long as there is a cycle in G_{i-1}, choose an edge e in any cycle of G_{i-1}, and form the subgraph G_i of G_{i-1} by deleting e from G_{i-1}.
- The final result G_k will be a spanning tree of G_0. This is a spanning tree.

Prim's Minimal Spanning Tree Algorithm

- Given a connected, simple graph G with $n+1$ nodes...
- Let v_0 be any node in G, and let $T_0 = \{v_0\}$ be a tree with one node and no edges.
- For each k from $\{1, 2, ..., n\}$...
  - Let $E_k = \{e$ an edge in G : e has one endpoint in T_{k-1} and the other endpoint not in T_{k-1}.
  - e_k be the edge in E_k with the smallest weight. (In case of a tie, choose any edge of the smallest weight.)
  - Let T_k be the tree obtained by adding edge e_k (along with its node not already in T_{k-1}) to T_{k-1}.
- T_n is the tree returned by the algorithm.

Closed walk: A walk where the beginning and end node are the same.

Length of a walk: The amount of edges that are in a walk.

Circuit: A closed trail

Cycle: A nontrivial circuit where the only repeated node is the first/last one.

Eulerian: A trail or circuit where every edge is traversed.

Tree: A tree is a connected simple graph that has no cycles. Vertices of degree 1 in a tree are called leaves of the tree.

Isomorphism: Simple graphs G and H are isomorphic if there is a one-to-one and onto function f from the nodes of G to the nodes of H such that $\{v, w\}$ is an edge of G if and only if $\{f(v), f(w)\}$ is an edge of H.

In other words: If they're basically the same graph, but with the nodes drawn in different locations. Nodes may have different names, but if you can rearrange the graphs to match each other, then they're isomorphic.

Planarity: A simple, connected graph is called planar if there is a way to draw it (on a plane) so that no edges cross.

Adjacency matrix Given a graph G with vertex set $V = \{v_1, v_2, ..., v_n\}$ and edge set E, we define the **adjacency matrix** of G as follows. The matrix M is an $n \times n$ array of natural numbers, which we imagine having rows and columns labelled as follows:

Columns $1, 2, ..., n$

Row 1 →

Row 2 →

Row 3 →

Row ...

Row n →

The entry in row i, column j (referred to as the (i, j) - entry of M or, more concisely, M_{ij}) is defined as

$$M_{ij} = \text{the number of edges connecting } v_i \text{ and } v_j \text{ in } G.$$