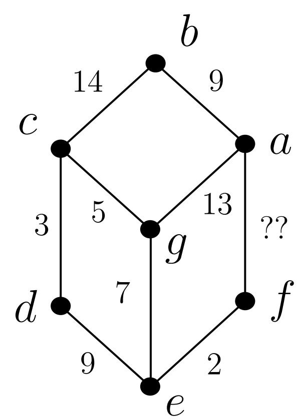
Math 101-002
Exam 2 Makeup, March 26

Name	CSU ID #

Be sure to read each question fully and carefully. Multiple choice answer bubbles must be fully filled in. There is space to the right of each multiple choice question to show work, if your work is correct you can get points even with an incorrect multiple choice answer.

1. For questions 1a through 1p consider the following weighted graph G:



(a) Write down the list of vertices of the graph G: (2 points)

$$V = \{\_\_\_, \_\_\_, \_\_\_, \_\_\_, \_\_\_, \_\_\_, \_\_\_\}\,.$$

(b) Write down the list of edges of the graph G: (2 points)

$$E = \{\_\_\_, \_\_\_, \_\_\_, \_\_\_, \_\_\_, \_\_\_, \_\_\_, \_\_\_, \_\_\_\}\,.$$

(c) Out of the following, which is **NOT** an edge in the graph? (2 points)

- ◯ bg
- ◯ cd
- ◯ eg
- ◯ af

(d) How many vertices does this graph have? (2 points)

- ◯ 3
- ◯ 6
- ◯ 7
- ◯ 8

(e) The degree of the vertex a is: (2 points)

$$\deg(a) = \_\_\_.$$

(f) The degree of the vertex e is: (2 points)

$$\deg(e) = \_\_\_.$$

(g) A path of length 5 from c to g passing through a is: (2 points)

- ◯ cg
- ◯ cb, ba, af, fe, ed
- ◯ cd, de, ef, fa, ag
- ◯ ga, ab, ba, ag, gc

(h) For the weight of the previous path to be 35, the missing weight should be: (2 points)

$$\text{weight} = \_\_\_.$$

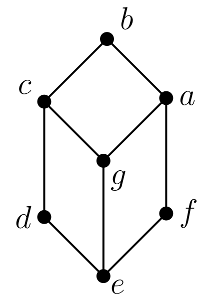
(i) Does this graph G have cut-edges (bridges)? (2 points)

- ◯ Yes.
- ◯ No.

(j) State whether this graph has an Euler walk. If it does, write it down, if not state why it doesn't. (2 points)

(k) State whether this graph has an Hamilton walk. If it does, write it down, if not state why it doesn't. (2 points)

(l) The graph G doesn't have an Euler circuit, Eulerize it by adding edges: (2 points)



(m) Assume you start traveling the graph G at the vertex f. If you are applying the Nearest-Neighbor algorithm and you traveled to the vertex a instead of e, then the weight of af was: (2 points)

- ◯ 1
- ◯ 2
- ◯ 3
- ◯ 4

(n) When traveling the graph G, you arrived at the vertex e from f. Following the Nearest-Neighbor algorithm, which vertex should you visit next? (2 points)

- ◯ a
- ◯ b
- ◯ d
- ◯ g

(o) When applying the Cheapest-Link algorithm, suppose you have already picked the edges ef, cd, cg and eg. Which edge should be picked next? (2 points)

- ◯ ab
- ◯ bc
- ◯ de
- ◯ ef

(p) Suppose you're applying Cheapest-Link algorithm to G. For af to be 3^{rd} edge you pick, its weight should be: (2 points)

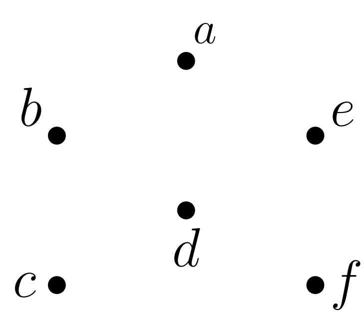
$$\text{weight}(af) = \_\_\_\_.$$

2. For questions 2a through 2k consider the information about the graph G given by the following lists:

$$V = \{a, b, c, d, e, f\},$$
$$E = \{ad, ae, bc, bd, cd, de, df, ea, ef\},$$
$$W = \{7, 6, 8, 9, 5, 6, 4, 10, 7\}.$$

The weights are ordered respecting the order of the edges.

(a) Fill in the weighted edges of the graph G (Hint: Pay attention to the repeated edge): (4 points)



(b) List all the vertices adjacent to d: (2 points)

$$N(d) = \{\_\_\_, \_\_\_, \_\_\_, \_\_\_, \_\_\_\}$$

(c) What are the degrees of the vertices a and f? (4 points)

$$\deg(a) = \_\_\_ \quad \text{and} \quad \deg(f) = \_\_\_.$$

(d) Find the sum of the degrees of all vertices: (2 points)

○ 6
○ 9
○ 12
○ 18

(e) State whether this graph has an Euler walk. If it does, write it down, if not state why it doesn't. (2 points)

(f) The graph G doesn't have an Euler circuit. Only one edge is needed to make it Eulerian. Which is that edge? (2 points)

$$\text{missing edge} = \_\_\_\_.$$

(g) Does this graph G have cut-edges (bridges)? (2 points)

- ◯ Yes.
- ◯ No.

(h) What is the weight of the path $\{bc, cd, df, fe\}$? (2 points)

- ◯ 5
- ◯ 8
- ◯ 20
- ◯ 24

(i) When traveling the graph G starting at e, your Nearest-Neighbor algorithm took you to a and then d. The next vertex you should visit is: (2 points)

- ◯ b
- ◯ c
- ◯ e
- ◯ f

(j) The Nearest-Neighbor algorithm starting at the vertex b produces a walk that ends at which vertex? (2 points)

- ◯ a
- ◯ b
- ◯ d
- ◯ f

(k) Which is the edge that you pick first in G when applying the Cheapest-Link algorithm? (2 points)

$$1^{\text{st}}\text{edge} = \_\_\_\_.$$

3. Short answer:

   (a) Explain the difference between Eulerian and Hamiltonian walks or circuits. (4 points)

   (b) Recall that a graph is called a *complete graph* when it contains all possible edges, is a graph that contains a *Hamilton circuit* complete? Briefly explain. (4 points)

4. In the following space draw the requested graphs. Answer briefly if requested.

   (a) A complete graph on 3 vertices. Does this graph have an Euler circuit? (4 points)

   (b) A complete graph on 4 vertices. Does this graph have an Euler circuit?. (4 points)

   (c) Draw a graph with an Euler circuit and a Hamilton walk, but not a Hamilton circuit. (4 points)

5. (25 points) In class, we've explored graph navigation algorithms to find Eulerian and Hamiltonian circuits. In real-world transportation networks, cities can often be approximated as **complete weighted graphs**, where locations are vertices and edges represent **travel times** between them.

   Suppose you are the **planning director** of a delivery company. Your fleet of trucks starts at a central warehouse and must deliver packages to various customer locations before returning. Consider the following:

   - Travel times change dynamically due to traffic (e.g., an edge weight may be **10 minutes in the morning but 35 minutes in the evening**).
   - Your goal is to **minimize total travel time** while ensuring every customer location is visited exactly once.
   - Deliveries may also have **time windows**, requiring some packages to be delivered by a specific time.

   In the next page thoroughly address and discuss the following points:

   - Should this delivery route be modeled as an **Eulerian or Hamiltonian circuit**? Justify your answer.
   - Compare the **nearest-neighbor algorithm** and **cheapest-link algorithm** for optimizing the route. Which would be better for **static planning** (i.e., setting routes in advance)?
   - How would your strategy change if deliveries had **time constraints**?
   - Given **real-time traffic updates**, is it beneficial to **dynamically switch between nearest-neighbor and cheapest-link**? Propose a hybrid approach.
   - Cities are only **approximated** as complete graphs. What are some **limitations** of this model? In what cases would a complete graph **not** be appropriate to make a model?