

Instructions: Please note that handwritten assignments **will not be graded**. Use the provided L^AT_EX template to complete your homework. Please do not alter the order or spacing of questions (keep each question on its own page). When you submit to Gradescope, you must mark which page(s) correspond to each question. **You may not receive credit for unmarked questions.**

When including graphical figures, we encourage the use of tools such as [graphviz](#) or packages like [tikz](#) for simple and complex figures. However, these may be handwritten only if they are neat and legible (as defined by the grader).

All answers must be concise and complete. See the Homework Guide for tips on how to write concise and complete answers.

1. For the following questions, select whether the statement is true or false, and write a *brief* explanation of your reasoning.

- (a) (4 points) [W2, ★] For undirected graphs, we really only need about half of an adjacency matrix to store the graph.

■ True □ False

This is because the matrix is symmetric and the diagonal is all 0, so we only need to store the upper or lower triangle.

- (b) (4 points) [W2, ★] Adjacency matrices allow one to quickly iterate through all the edges of a particular node, while adjacency lists allow us to check in constant time if two nodes are adjacent.

□ True ■ False

Adjacency matrices are constant time to check if two nodes are connected. Adjacency lists are linear time to iterate through list of connected nodes too check if two nodes are adjacent.

- (c) (4 points) [W2, ★] Adjacency lists use memory much more efficiently than adjacency matrices for sparse graphs.

■ True □ False

Adjacency lists only store the edges that exist, while adjacency matrices store all possible edges (including non-existing connections).

- (d) (4 points) [W2, ★] A BFS uses a queue data structure.

■ True □ False

No explanation necessary for this problem.

2. The following questions are related to the maze project. No explanation is necessary for these problems.

(a) (5 points) [W2, ★] Select all the statements that are true.

■ If Captain Rocket is in a room colored B, Lieutenant Lucky can move along a corridor which is colored B outgoing from the room Lieutenant Lucky is currently in.

□ Captain Rocket and Lieutenant Lucky must alternate and each should move every-other turn.

□ Both Captain Rocket and Lieutenant Lucky must get to the Goal to solve the maze.

■ Each corridor has a single direction along which travel is possible.

(b) (2 points) [W2, ★] Select the true statement.

□ The graph traversal algorithm used in my project should be DFS.

■ The graph traversal algorithm used in my project should be BFS.

□ I need to create my own graph traversal algorithm to complete this project.

(c) (2 points) [W2, ★] In the input for the project, if you have the following line

4 6 A

then there is a corridor from room 4 to room 6 with color A.

■ True □ False

(d) (5 points) [W2, ★] If the first three lines of the input file are

10 20

G R B R G B B R G R

4 7

select all the statements that are true.

■ There are 10 rooms and 20 corridors.

□ There are 20 rooms and 10 corridors.

□ Captain Rocket starts in room 7 and Lieutenant Lucky starts in room 4.

■ Captain Rocket starts in room 4 and Lieutenant Lucky starts in room 7.

■ There are three distinct colors: G, R, and B.

3. (10 points) [W2, ★★] While working at a large Fortune 500 company, your CEO boasts “even though our company is large, there are no more than 7 levels between me and any employee.” Given the organizational hierarchy of the company, how can you verify the CEO’s claim? (Your answer should be a few sentences at most.)

This can be verified by:

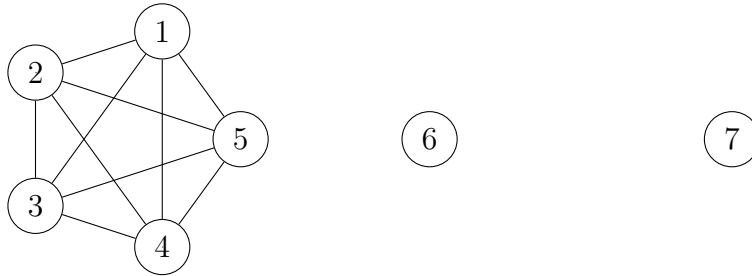
- Use BFS to find the shortest path between the CEO and any employee.
- If any path is greater than 7, then the CEO’s claim is false.

4. (10 points) [W2, ★★★] Connected Components.

- (a) (5 points) Is it possible to draw an undirected, unweighted, simple graph with exactly 7 vertices, **11** edges, and 3 connected components? If so, then give the graph. If not, explain why it is not possible.

No this is not possible. To achieve the 3 we need 2 vertices with no edges, and a complete graph with 5 vertices. This would give us 10 edges, so it is not possible. (See next question for example)

- (b) (5 points) Is it possible to draw an undirected, unweighted, simple graph with exactly 7 vertices, **10** edges, and 3 connected components? If so, then give the graph. If not, explain why it is not possible.



5. (20 points) [W2, ★★★] Graph Coloring. Remember from class that a graph coloring is an assignment of colors to vertices such that no edge connects two vertices of the same color.

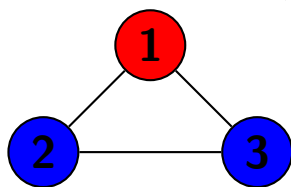
A **complete graph** is a graph in which every node has an edge to every other node.

- (a) Is it possible to two-color a complete graph of arbitrary size?

☐ Yes ☒ No

- (b) Provide a *concise but complete* proof for your answer to (a).

Counterexample: Complete graph of size 3 cannot be two-colored. If we color the first node red, then the other two nodes must be blue. However, the other two nodes are connected, so they cannot be the same color.



- (c) If it is not possible to two-color a complete graph of arbitrary size, then what is the minimum number of colors required to color a complete graph with v vertices? Provide a *concise but complete* proof. If it is possible to two-color a complete graph of arbitrary size, leave this question blank.

Claim: For a complete graph K_v with v vertices, the required colors $\chi(K_v)$ is equal to the number of vertices v .

Proof:

Base Case: For $v = 1$, the complete graph K_1 consists of a single vertex, and it trivially requires only one color. Thus, $\chi(K_1) = 1$.

Inductive Step: Assume that for a complete graph K_n , the chromatic number $\chi(K_n)$ is n . Now, consider adding a new vertex to form K_{n+1} .

- The new vertex is adjacent to all n existing vertices in K_n .
- To color K_{n+1} , we need a new color for the newly added vertex since it is adjacent to all existing vertices.
- Therefore, $\chi(K_{n+1}) = n + 1$.

By induction, we have shown that the required colors of a complete graph with v vertices (K_v) is equal to the number of vertices v , i.e., $\chi(K_v) = v$. Thus, the minimum number of colors required to color a complete graph with v vertices is v .

6. (30 points) [W2, ★★☆☆] Graph Modelling. Consider the following problem.

University Scheduling

Consider a simplified university where each student takes exactly two courses. There are two time slots—morning and afternoon (that is, the student will have their morning course every morning, and their afternoon course every afternoon). Given the courses that each student has signed up for, determine if it is possible to teach each course exactly once (either in the morning or in the afternoon but not both) while satisfying all of the students course requirements.

Provide a concise but complete description of how to model this problem as a graph and use an algorithm we have discussed in class to determine the answer. Be sure to describe how to interpret the result of the chosen algorithm to produce an answer to the problem.

Note: It is recommended to provide the answer to this question in a list of steps.

Create Graph:

1. Create a vertex for each course.
2. Create an edge for each student, connecting the two courses they are taking.
3. Assign each vertex a binary color (morning or afternoon).

Algorithm: Use DFS

1. Start at any vertex.
2. Assign the vertex a color.
3. Assign the adjacent vertices the opposite color.
4. Repeat for all unvisited vertices.
5. If a vertex is assigned a color that is the same as an adjacent vertex, then the graph is not 2-colorable.
6. If all vertices are assigned a color, then the graph is 2-colorable.

Interpretation: If the graph is 2-colorable, then it is possible to teach each course exactly once while satisfying all of the students course requirements. Otherwise, it is not possible.