CS141, Winter 2025, Assignment #5 **Due: 11:59 PM, 3/15/2025**
Name: **Net ID:**

Deadline. The homework is due 11:59 PM, 3/4/2025. You must submit your solutions (in pdf format generated by LaTeX) via GradeScope. Canvas will not be accepted.
Assignment Policy.

- Unless mentioned otherwise, late submission allowed for 20% penalty for each calendar day.
- Assignments should be in pdf format generated by LaTeX
- If you are using any external source, you must cite it and clarify what exactly got out of it.
- You are expected to understand any source you use and solve problems in your own.
- All pages relevant to a question must be assigned on Gradescope. Unassigned pages will not be graded.
- Unless stated in the question, your designed algorithm should use the most optimized algorithm you have learned so far. A too-complex algorithm may result in penalties.

1 Reverse Delete (20 points)

Consider the following algorithm for MST, called **Reverse-Delete**. Start with the full graph $G = (V, E)$ and begin deleting edges in order of decreasing cost. As we get to each edge e (starting from the most expensive), we delete it as long as doing so would not actually disconnect the graph we currently have.

Prove that **Reverse-Delete** computes the optimal MST. As this is a greedy algorithm, prove the greedy choice and the optimal substructure.

2 Water Transport (20 points)

You are given three jugs that have the following capacities (in pints): 10, 7, and 4. The 7-pint and 4-pint jugs start out full of water, and the 10-pint jug is initially empty. We are allowed only one operation: Pour the water from one jug to another until either 1) the source jug is empty or 2) the destination jug is full.

We want to know if there is a sequence of pours such that the 7-pint jug or 4-pint jug has exactly 2 pints of water.

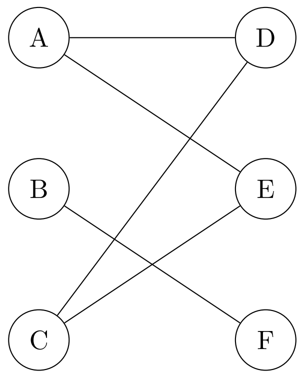
1. Model this problem as a graph problem: That is, model this problem with nodes and edges, and define what each node and edge represents.
2. Once this problem is modeled as a graph, what algorithm can we use to either find a solution or determine that no solution exists?
3. Is there a solution? If so, describe the sequence of pours to reach that solution. If not, state why you know there is no solution.

3 Huffman Encoding (20 points)

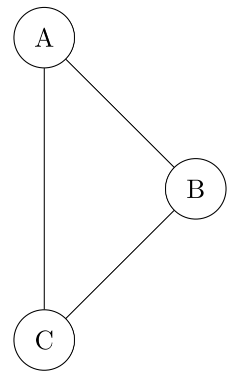
The Huffman encoding creates a binary tree of nodes where each node represents a character and the weight of the node is the frequency of the character. The tree is constructed by repeatedly combining the two nodes with the smallest weight into a new node with the sum of the weights (we call these synthetic nodes). The tree is traversed to create the encoding of each character.
Given the string "abracadabra", create the Huffman encoding tree. What is the code length?

4 Bipartite Graphs (20 points)

Bipartite



Non-Bipartite



A bipartite graph is a graph whose set of vertices can be divided into two disjoint and independent sets such that no two vertices within the same set are adjacent. We can check if a graph is bipartite by attempting to color the graph with two colors in a way that no two adjacent nodes have the same color. If we can do this, the graph is bipartite (you can prove this to yourself by shading the examples above).

We would like to write an algorithm that determines if a graph is bipartite along with the set assignments. Write a BFS-based algorithm that determines if a graph is bipartite. You may assume that the graph is connected, and represented as an adjacncy list. Assume set operations can be done in $O(1)$ time.

Note: We don't know which nodes are in which set to begin with, in other words, we're not verifying that an assignment is correct. Instead, we're trying to find an assignment that works. Your algorithm should return which nodes are in the first, or second set.

5 Infrastructure (10 point bonus)

You are given a set of cities, along with the pattern of highways between them in the form of an undirected graph $G = (V, E)$. Each stretch of highway $e \in E$ connects two of the cities, and you know its length in miles, l_e. You want to get from city s to city t. There is one problem: your car can only hold enough gas to cover L miles. There are gas stations in each city, but not between cities. Therefore, you can only take a route (path) if every one of its edges has length $l_e < L$.

1. Given the limitation of your car's fuel tank capacity, show how to determine in linear time whether there is a feasible route from s to t.

2. You are planning to buy a new car, and you want to know the minimum rule tank capacity that is needed to travel from s to t. Give a $O((n + m) \log n)$ algorithm to determine this.

Hint for (2): Modify Dijkstra's algorithm to find paths that minimize the maximum weight of any edge on the path (instead of the path length).

6 Boolean Matrix (10 point bonus)

You are given an $m \times n$ Boolean matrix, M. If $M_{i,j} = 1$, that represents a pixel that is part of an image. The full image is all adjacent 1's in M. Adjacent means directly above, below, left, or right (diagonal is not adjacent). If $M_{i,j} = 0$, then that position does not contain a pixel that is part of an image. We have provided a sample matrix with sample solutions for clarity.

$$M_s = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

1. Provide an $O(mn)$ algorithm that determines the largest image (most adjacent pixels) in M. In M_s, the solution is 4 (the top left corner of the matrix).
2. Provide an $O(mn)$ algorithm that determines the number of different images in M. In M_s, the solution is 5. (top left corner, top right corner, bottom right corner, bottom middle, and $M_s(2,4)$ (1-based index)).

7 Two Parts (10 point bonus)

For the boolean matrix problem, $O(mn)$ is an optimal solution, however there are multiple ways to implement an $O(mn)$ algorithm.

1. Provide another $O(mn)$ algorithm for 4.2 (+5 points)
2. Suppose we redefine adjacent such that diagonal 1's are now also considered adjacent. Modify your algorithm from 4.2 to accommodate this new definition. How many images are in M_s under this new definition? (+5 points)