COMS 3110: Homework 4
Due: July 29^{th}, 11:59pm
Total Points: 70

Submission format. Homework solutions must be typed. You can use word, LaTeX, or any other type-setting tool to type your solution. Your submission file should be in pdf format. Do **NOT** submit a photocopy of handwritten homework except for diagrams that can be hand-drawn and scanned. We reserve the right **NOT** to grade homework that does not follow the formatting requirements. Name your submission file: `<Your-net-id>-3110-hw4.pdf`. For instance, if your netid is `asterix`, then your submission file will be named `asterix-3110-hw4.pdf`. Each student must hand in their own assignment. If you discussed the homework or solutions with others, a list of collaborators must be included with each submission. Each of the collaborators has to write the solutions in their own words (copies are not allowed).

General Requirements

- When proofs are required, do your best to make them both clear and rigorous. Even when proofs are not required, you should justify your answers and explain your work.
- When asked to present a construction, you should show the correctness of the construction.

Some Useful (in)equalities

- $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$
- $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$
- $2^{\log_2 n} = n$, $a^{\log_b n} = n^{\log_b a}$, $n^{n/2} \leq n! \leq n^n$, $\log x^a = a \log x$
- $\log(a \times b) = \log a + \log b$, $\log(a/b) = \log a - \log b$
- $a + ar + ar^2 + ... + ar^{n-1} = \frac{a(r^n-1)}{r-1}$
- $1 + \frac{1}{2} + \frac{1}{2^2} + ... + \frac{1}{2^n} = 2(1 - \frac{1}{2^{n+1}})$
- $1 + 2 + 4 + ... + 2^n = 2^{n+1} - 1$

---

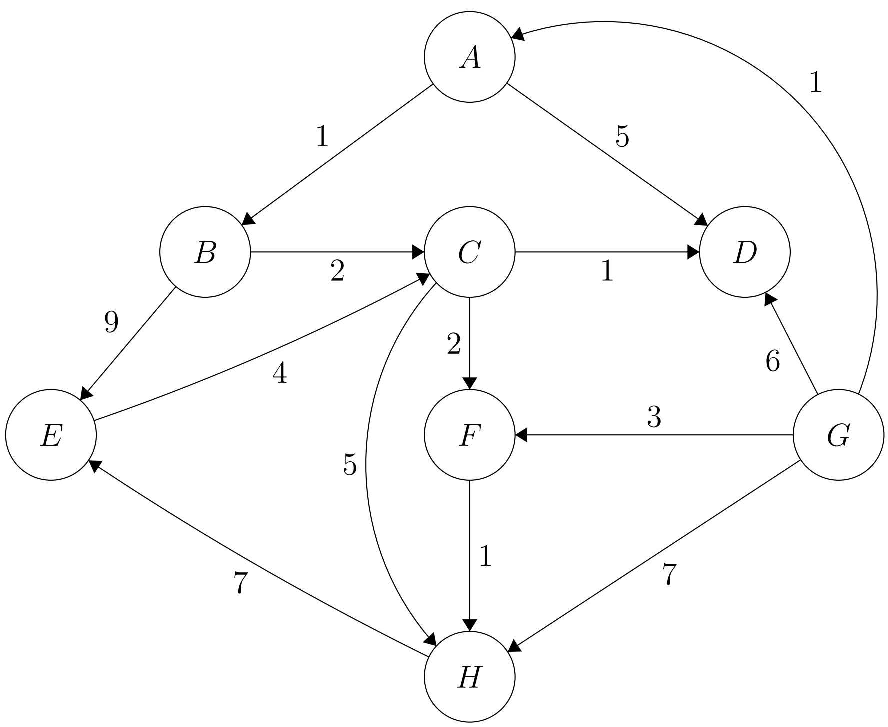
1. **(20 points)** Given an unweighted undirected graph $G = (V, E)$ as an adjacency list and two vertices v_1, v_2 in V, design an efficient algorithm to determine the number of shortest paths from v_1 to v_2. Analyze the runtime of your algorithm.

2. **(20 points)** Design an efficient algorithm that, given a directed graph $G = (V, E)$ represented as an adjacency list, determines if there exists a vertex $v \in V$ with the following property: $\forall w \in V, w \neq v$, there exists a path from w to v in the graph. It should return true if so and false otherwise. Analyze the runtime of your algorithm.

3. **(15 points)** Given an undirected graph $G = (V, E)$, we say that it is k-colorable if each vertex $v \in V$ can be colored with one of the k colors such that $\forall (v_1, v_2) \in E : c(v_1) \neq c(v_2)$, where $c(v)$ is the color of vertex v.

   Design an efficient algorithm that determines if the given graph is 2-colorable. It must take as input a graph G and two colors c_1, c_2. If the graph is not 2-colorable, it should return *nil*. If the graph is 2-colorable, a valid color assignment for each vertex should be returned. Analyze the runtime of your algorithm.

4. **(15 points)** Consider Dijkstra's shortest path algorithm (as described in class) ran on the following graph starting at vertex A. Determine the d-value for each vertex at the end of each iteration of the outer while loop in the algorithm.



(You may add or remove rows to the following table, if needed, for your solution.)

Initial d-value	0	∞	∞	∞	∞	∞	∞	∞
Iteration	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6								
7								
8								