In preparing your solutions to the exam, you are **allowed to use any sources** including textbooks, other students and professors, previous homeworks and solutions, or any sources on the Internet. The only source you may *not* use is me. Of course, I am happy to answer general questions, go over homework problems, or answer clarifying questions about exam problems.

The exam will take place in class on Friday, October 4 (MC Reynolds 317, 1:10–2:00pm). You are **not** allowed to bring any notes, textbooks, calculators, or any other resources with you to the exam. Bring only something to write with; I will provide a fresh copy of the exam, paper for writing your solutions, and scratch paper.

Question 1 (24 points (4 points each)). Characterize the asymptotic behavior of each of the following in terms of Θ, as a function of n and/or m. Give a short proof/justification for each answer. Full credit will only be given for the best (fastest) possible algorithms.

(a) $1 + 2 + 3 + \cdots + n/2$

(b) $4 + 8 + 16 + 32 + \cdots + 2^n$

(c) Time to find the shortest path between two given vertices in an unweighted, undirected graph with n vertices and m edges.

(d) Time to find the shortest path between two given vertices in a weighted, undirected graph (assuming all weights are nonnegative) with n vertices and m edges.

(e) Given an array A of size n, the time to fill in a matrix M such that $M[i, j] = A[i] + A[j]$.

(f) Given an array A containing n positive integers, the time to compute the smallest positive integer which is not contained in the array.

Question 2 (20 points). Prove, or disprove with a counterexample: for all $k \geq 1$, if a tree T has a vertex of degree k, then T has at least k leaves.

Remember that by *tree* we mean a tree in the graph-theory sense, not a tree data structure with a root and children and so on.

Question 3 (20 points). Consider the following algorithm to determine whether an undirected, unweighted graph G has any cycles: pick an arbitrary vertex v and run a breadth-first search (BFS), generating a sequence of layers $L_0, L_1, L_2, \ldots$. If there is any edge between two vertices in the same layer, then G has a cycle; otherwise, G has no cycles.

Prove or disprove the correctness of this algorithm.

Question 4 (25 points)**.** Recall *Dijkstra's algorithm* for finding shortest paths in a directed, weighted graph.

(a) (10 points) Why doesn't Dijkstra's algorithm work if edges in the graph can have negative weights (even if there are no cycles)? Give an example of a directed acyclic graph where Dijkstra's algorithm fails to find the minimum-weight path between a pair of vertices. Be sure to demonstrate that you understand *why* your graph is a counterexample; that is, show what Dijkstra's algorithm does on your example graph, and explain why the path it finds is not the minimum-weight path.

(b) (15 points) What happens if we replace the word "smallest" in Dijkstra's algorithm with the word "biggest"—that is, we use a max priority queue so we pull out the vertex u with the *maximum* distance on each iteration, and for each edge (u, v) we update $d[v]$ to be the *maximum* of $d[v]$ and the sum $d[u] + w_{uv}$. Can we use this modified Dijkstra's algorithm to find *longest* paths bewteen nodes in a graph? Prove that this works, or give a counterexample (with explanation) where it doesn't.

Question 5 (36 points)**. Describe** (13 points) a $\Theta(V + E)$ algorithm which, given a directed graph $G = (V, E)$ as input, will either find a directed cycle in G, or report that G is acyclic. You should describe your algorithm using appropriate pseudocode, at a level of detail that would enable someone else to turn your algorithm into working code (similar to what we have done in class). **Prove** (13 points) your algorithm is correct, and **analyze** (10 points) its asymptotic running time.