Remember that you must write up your solutions **independently**, and **list your collaborators** by name clearly at top of your submission (or "no collaborators" if none).

76 points total, with an additional 12 bonus points.

1. (12 pts total: 3pts each) Some probability boot camp:

a. Prove Markov's inequality: for any nonnegative random variable X and $c > 0$, $\Pr[X \geq c] \leq \mathbb{E}[X]/c$. (Hint: I'll give you the proof:

$$\Pr[X \geq c] = \mathbb{E}[\mathbb{1}\{X \geq c\}] = \frac{1}{c}\,\mathbb{E}[c\mathbb{1}\{X \geq c\}] \leq \frac{1}{c}\,\mathbb{E}[X]\ ,$$

where $\mathbb{1}\{E\}$ is the indicator function for event E, taking value 1 in E and 0 otherwise. All you have to do is justify each of these steps.)

b. Prove Chebyshev's inequality: for any random variable with mean μ and variance σ^2, $\Pr[|X - \mu| \geq c \cdot \sigma] \leq 1/c^2$. (Hint: use Markov.)

c. Look up the definition of conditional probability and conditional expectation, and show that for any discrete random variables X, X', $\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X|X']]$.

d. A *martingale* is a sequence of random variables $X_0, X_1, X_2, \ldots$ such that $\mathbb{E}[X_{t+1}|X_0, \ldots, X_t] = X_t$ and $X_0 = 0$. Prove by induction that $\mathbb{E}[X_t] = 0$. You may assume (**c**) holds for arbitrary random variables (as it does).

2. (6 pts) Give an example of a directed graph $G = (V, E)$, a source vertex $s \in V$ and a set of tree edges $E_\pi \subseteq E$ such that for each vertex $v \in V$, the unique path in the graph (V, E_π) from s to v is a shortest path in G, yet the set of edges E_π cannot be produced by running a breadth-first search on G, no matter how the vertices are ordered in each adjacency list. Include a paragraph explaining why your example works.

3. (6 pts) A *cut* in a graph $G = (V, E)$ is a partition (S, S') of its vertices, i.e. $S \cap S' = \emptyset$ and $S \cup S' = V$. Give an efficient algorithm for the following problem.

Input:	Undirected weighted graph $G = (V, E)$, edge $e \in E$
Output:	A cut such that e is a lightest edge across the cut, or return NO if no such cut exists

4. (10 pts) Often when an adjacency matrix representation is used, most graph algorithms take $\Omega(|V|^2)$ time, but there are some exceptions. Given a directed graph G, a *universal sink* is defined as a vertex with in-degree $|V| - 1$ and out-degree 0 (i.e., every other vertex points to this one). Give a $o(|V|^2)$ algorithm to solve the following problem. The more efficient your algorithm is, the more credit you will get.

Input:	Adjacency matrix M of a directed graph G (possibly with self-loops)
Output:	A universal sink of G, or return NO if none exists

5. (8 pts total) A graph (V, E) is *bipartite* if the vertices V can be partitioned into two subsets L and R, such that every edge has one end in L and the other in R.

a. (3 pts) Prove that every tree is a bipartite graph.

b. (5 pts) Adapt an algorithm described in class so that it will efficiently determine whether a given undirected graph is bipartite.

6. (8 pts total, **plus 8 total bonus pts**) In a late-night algorithms study session, you and Golum argue about the conditions under which a minimum spanning tree is unique. You agree that if all edges in G have unique weights the MST is also unique, but you disagree about how to relax this assumption. Let $w(e)$ be a function that returns the weight of some $e \in E$.

a. (4 pts) Give a small example graph with both a unique MST and two edges of equal weight. Give a recipe to construct a graph with $\Omega(|V|^2)$ edges of equal weight yet still having a unique MST.

b. (4 pts) Golum claims the following statement. Find a small examples refuting it.
Golum's Claim: G has a unique MST if and only if (i) every cut has a unique minimum-weight edge across it, and (ii) the maximum-weight edge in any cycle of G is unique.

c. **(Optional: 4 bonus pts)** Golum now demands that you produce the correct relaxed condition, which you claim is the following. Prove that you are correct.
Your Claim: a weighted graph G has a unique *MST T_{mst} if and only if the following conditions hold:*
(i) any cut induced by removing some edge $e \in T_{\text{mst}}$ (the remaining two components of T_{mst} form the cut in G) has a unique minimum-weight edge crossing it,
(ii) the maximum-weight edge of any cycle constructed by adding one edge e' to T_{mst}, where $e' \notin T_{\text{mst}}$, is unique.
Hint: Note that for any spanning tree T on G, removing some edge $e \in T$ induces a bi-partition of the vertices. Consider the edges that span this cut.

d. **(Optional: 4 bonus pts)** Describe and analyze an algorithm that will determine whether an input graph G has a unique MST in (effectively) $O(|E|\log|V|)$ time.

7. (10 pts) Suppose there is a jogging path, which is represented as the number line. There are n joggers who use the path, and for each jogger you know what segment of the path they use, which is an interval $[a_i, b_i]$ where $a_i < b_i$. You want to put up billboards advertising your business. You can place billboards anywhere you want along the jogging path, but you wish to minimize the number of billboards subject to the constraint that every jogger sees at least one of your billboards. Give an $O(n \log n)$ time greedy algorithm for this problem.

Input:	Intervals $[a_i, b_i]$ where $a_i < b_i$, for $i = 1, \ldots, n$
Output:	Set of numbers S such that $\forall i\ \exists x \in S : a_i \leq x \leq b_i$ and $|S|$ is minimized

8. (10 pts) This is a continuation of the previous problem. Again there is a jogging path, which is represented as the number line. There are n joggers who use the path, and now each jogger uses two segments of the path. In other words, there are two intervals $[a_i, b_i]$ and $[c_i, d_i]$ where $a_i < b_i < c_i < d_i$ such that the ith jogger's route uses the segment $[a_i, b_i]$, then leaves the jogging path for a while, and then returns to use the segment $[c_i, d_i]$. Again you wish to place billboards along the jogging path, minimizing the number of billboards subject to the constraint that every jogger sees at least one of your billboards. Prove that the search version of this problem is NP-complete.

Input:	Intervals $[a_i, b_i]$ and $[c_i, d_i]$ where $a_i < b_i < c_i < d_i$, for $i = 1, \ldots, n$, and an integer M
Output:	Set of numbers S such that $\forall i\ \exists x \in S : (a_i \leq x_j \leq b_i$ or $c_i \leq x_j \leq d_i)$ and $|S| \leq M$, or return NO if no such S exists

9. (6 pts) Suppose you have a strongly connected directed graph with positive edge lengths. Someone gives you a number d_v for each node v and claims that these are the shortest path distances from a particular source node s. However, you distrust the person and would like to verify this claim. One way to do it is to run Dijkstra's algorithm and compute the correct distances from scratch, but you realize there might be a better way. Give a linear-time algorithm for checking whether the d_v's are the correct distances.

Input:	Strongly connected directed graph $G = (V, E)$ with edge lengths $\ell_e > 0$ (for $e \in E$), node $s \in V$, numbers d_v (for $v \in V$)
Output:	YES or NO: $(\forall v \in V\ :\ d_v$ is the shortest path distance from s to $v)$?

10. (Optional: 4 bonus pts) Suppose you have implemented the disjoint-sets data structure using union-by-rank but not path compression. Give a sequence of m `union` and `find` operations on n elements which together take $\Omega(m \min(\log n, \log m))$ time.