This review section covers material up to and including dynamic programming (Quiz 2).

**Problem 1. True or False**

Answer True or False to the following questions:

(a) Every non-empty DAG has at least one source.

(b) Suppose we have a graph where each edge weight value appears at most twice. Then, there are at most two minimum spanning trees in this graph.

(c) A dynamic programming problem where we need to solve \(O(n^2)\) subproblems will take \(O(n^2)\) time.

**Problem 2. Big O Only**

Is it possible to find two functions \(f(n), g(n)\) such that \(f(n) = O(g(n))\), but \(f(n) \neq o(g(n))\), \(f(n) \neq \Omega(g(n))\)? If so, give an example. If not, give a proof.

**Problem 3. Cyclic Sorted Array**

You’re given an array \(A\) of \(n\) distinct numbers that is sorted up to a cyclic shift, i.e. \(A = (a_1, a_2, \ldots, a_n)\) where there exists a \(k\) such that \(a_k < a_{k+1} < \ldots < a_n < a_1 < \ldots < a_{k-1}\), but you don’t know what \(k\) is. Give an efficient algorithm to find the largest number in this array.
Problem 4. Counting Permutations
How many permutations of \{1, 2, \ldots, N\} have exactly \(K\) inversions? An inversion is a pair of numbers \(i, j\) in the permutation such that \(i > j\) but \(i\) comes before \(j\). For example, for \(N = 3\) and \(K = 1\) the answer is 2: \{2, 1, 3\} and \{1, 3, 2\}.

Problem 5. Dijkstra’s Variations
In lecture, we showed that Dijkstra’s algorithm may not work correctly if negative edge weights are present. In this problem. Let \(G = (V, E)\) be a weighted, directed graph with no negative-weight cycles. For each of the following property of \(G\), give an algorithm that finds the length the shortest path from \(s\) to all other vertices in \(V\). In both cases, your algorithm should have the same run-time as that of Dijkstra’s.

(a) \(G\) only has one negative edge. Give an algorithm to find length of the shortest path from \(s\) to all other vertices in \(V\).

(b) The edges leaving from the source \(s\) may have negative weights, but all other edges in \(E\) have non-negative weights,
Problem 6. Largest Submatrix of 1’s
You are given a matrix $M$ with $m$ rows and $n$ columns consisting of only 0’s and 1’s. Give an efficient algorithm for find the maximum size square sub-matrix consisting of only 1’s.

Problem 7. Minimum Bottleneck Spanning Tree
Let $G = (V, E)$ be an undirected graph with weights $w_e$ on the edges. A minimum bottleneck spanning tree of $G$ is a spanning tree such that the weight of the heaviest edge is as small as possible. Note that for minimum bottleneck spanning trees, we do not care about the sum of the weights of the chosen edges—only the maximum edge.

(a) Show that any minimum spanning tree is necessarily a minimum bottleneck spanning tree

(b) Give an example of a graph and a minimum bottleneck spanning tree that is not a minimum spanning tree.
Problem 8. Well-Nested Brackets
There are four types of brackets: (, ), <, and >. We define what it means for a string made up of these four characters to be well-nested in the following way:

(a) The empty string is well-nested.
(b) If A is well-nested, then so are <A> and (A).
(c) If S, T are both well-nested, then so is their concatenation ST.

For example, (), <>, (()) , ()<>, and ()<> are all well-nested. Meanwhile, (, <, ), )(, (<)>, and <(> are not well-nested.

Devise an algorithm that takes as input a string s of length n made up of these four types of characters. The output should be the length of the shortest well-nested string that contains s as a subsequence (not sub-string). For example, if <(> is the input, then the answer is 6; a shortest string containing s as a subsequence is ()<>().

Problem 9. Efficient Waiters
A restaurant has n tables and 2 waiters. Table i is located at coordinate $(x_i, y_i)$ in the restaurant for $i = 1, 2, \ldots, n$. At the beginning, one waiter is at table 1 and the other waiter is at table 2. We will then receive a sequence of m queries. Each query is simply a number in $\{1, 2, 3, \ldots, n\}$ representing that the customer at that table needs some kind of service. Hence, one of the two waiters needs to be service that table. The waiters are complaining of walking too much and so the restaurant has hired you to help them be more efficient.

Suppose an omniscient being has told you in advance the order in which the tables would need servicing. Let $t_1, t_2, \ldots, t_m$ be the sequence of queries (i.e. tables that need servicing). The goal is to minimize the total walking distance of the two waiters, where the distance between two tables is the standard Euclidean distance. Assume that there is enough time between the queries for whichever of the 2 waiters is chosen to walk to the table and service it.

(a) Consider the following greedy algorithm. In order to service table $t_i$, we will have whichever waiter is closer to service that table. Give an example to show that this greedy algorithm is not optimal.
(b) Give a $O(n^2 m)$ time and $O(n^2)$ space solution that finds the value of the shortest total walking distance.

Problem 10. Significant Inversions (Kleinberg-Tardos 5.2)

Call a significant inversion in an array a pair $i < j$ such that $a_i > 2a_j$. Give an $O(n \log n)$ algorithm to count the number of significant inversions in an array.