CS 473: Algorithms, Spring 2021
HW 3 (due Wednesday, February 24th at 8pm)

This homework contains three problems. **Read the instructions for submitting homework on the course webpage**.

Collaboration Policy: For this home work, each student can work in a group with up to three members. Only one solution for each group needs to be submitted. Follow the submission instructions carefully.

1. **[10pts]** You are part of a team that competes in a challenge to design a robot with the purpose of traversing a rectangular grid of M rows and N columns. Your team positions the robot at $(1, 1)$, which is the top-left cell. The goal is for the robot to reach (M, N), i.e. the bottom-right cell. In a single step, the robot can move only to the cells to its immediate right and bottom directions. More specifically, if the robot is at position (i, j), then it can move either to position $(i + 1, j)$ or to $(i, j + 1)$, provided that both positions are inside the grid.

   The challenge is that there are P obstacles in different positions on the grid, through which the robot cannot pass. Given the positions of the blocked cells, your task is to count all the number of paths that the robot can take to move from $(1, 1)$ to (M, N).

   A solution whose running time depends only on N and M will only get partial credit. We want a solution which is fast when P is small.

2. **[10pts]** You are given a directed graph $G = (V, E)$ where each edge e has a length/cost c_e and you want to find shortest path distances from a given node s to all the nodes in V. Suppose there are only k edges $f_1 = (u_1, v_1), f_2 = (u_2, v_2), \ldots, f_k = (u_k, v_k)$ that have negative length, and the rest have non-negative lengths. Here think k is small, say a constant number. The Bellman-Ford algorithm for shortest paths with negative length edges takes $O(nm)$ time where $n = |V|$ and $m = |E|$. Show that you can take advantage of the fact that there are only k negative length edges to find shortest path distances from s in $O(kn \log n + km)$ time — effectively this is the running time for running Dijkstra's algorithm k times. Your algorithm should output the following: *either* that the graph has a negative length cycle reachable from s, *or* the shortest path distances from s to all the nodes $v \in V$.

   Hint: First solve the case when there is only a single negative length edge. You will get half the credit if you only solve this case. Then solve the case of two negative length edges and see if you can generalize that approach. Note that this is only one approach, there may be others.

3. In the selection problem we are given an array A of n numbers (not necessarily sorted) and an integer k, and the goal is to output the rank k element of A. Consider a randomized algorithm where we pick a number x uniformly at random from A and use it as a pivot as in quick sort to partition A into numbers less than equal to x and numbers greater than x. The algorithm recurses on one of these arrays depending on k and the size of the two arrays. It can be shown that this algorithm runs in $O(n)$ expected time and has the advantage of being quite simple when compared to the median of median (deterministic) algorithm.

(A) **[5pts]** Write down a description of randomized quick selection in pseudocode. Show that the expected depth of the recursion of randomized quick selection is $O(\log n)$. (You can also prove that the expected running time is $O(n)$ but you should prove this for yourself since it will help you in the next part. However, you don't have to submit it as part of the home work.)

Hint: Write a recurrence for the depth of the recursion.

(B) **[5pts]** Let $A_1, A_2, \ldots, A_h$ be h *sorted* arrays where A_i has n_i elements. Let $n = \sum_{i=1}^{h} n_i$. Assume that the arrays have distinct elements. Describe a randomized algorithm that given integer k finds the k'th smallest element in the combined set of arrays in $O(h \log^2 n)$ expected time.

Hint: Adapt the randomized quick selection algorithm and the analysis from the first part.