Due Friday Oct. 1 at 11:59 p.m.

1. (a) (5 + 5+ 5 points) Decide whether each of the following is true or false. If it is true, give a short explanation. If it is false, give a counterexample.

   i. Suppose we are given an instance of the Shortest $s-t$ Path Problem on a directed graph G. We assume that all edge costs are positive and distinct. Let P be a minimum-cost $s-t$ path for this instance. Now suppose we replace each edge cost c_e by its square, c_e^2, thereby creating a new instance of the problem with the same graph but different costs. True or false? P must still be a minimum-cost $s-t$ path for this new instance.

   ii. Suppose we are given an instance of the Minimum Spanning Tree Problem on a graph G, with edge costs that are all positive and distinct. Let T be a minimum spanning tree for this instance. Now suppose we replace each edge cost c_e by its square, c_e^2, thereby creating a new instance of the problem with the same graph but different costs. True or false? T must still be a minimum spanning tree for this new instance.

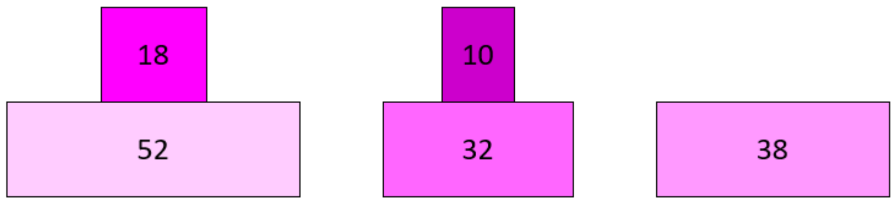
   iii. Consider the Interval Scheduling Problem from class. We know that we can obtain the optimal solution by scheduling jobs in increasing order of finish times. Say you are given a schedule R, and you've found an optimal solution for the same, let us call this optimal solution as S. Now suppose that the interval durations of each of the jobs is squared (we do this by extending the finish times of each job), let us call the resulting schedule as R'. Would the optimal solution S for the schedule R still be an optimal solution for the new schedule R'?

   (b) (20 points) Alex is playing a video game in which he is in a big maze filled with m passages (edges) denoted by $[1,\ldots,m]$. He has to get from the entrance to the exit using these passages. However, each such passage has a possibility of disappearing before the start of the game, in which case the passage can no longer be used by Alex during the game. For each passage i we associate a number $p_i \in [0,1]$ which denotes the probability that the passage will *not* disappear during the game. The layout of the maze and these passage probabilities are known to Alex at the start of the game, help Alex by finding the most promising path i.e. a path from the entrance to the exit with the highest probability of not disappearing. Note that once the game is started, the passages no longer change/disappear.
   Hint: Let a path from entrance to exit consist of the passages $(2,4,m-1)$. Then, probability that the path does not disappear during the game is $p_2 p_4 p_{m-1}$.

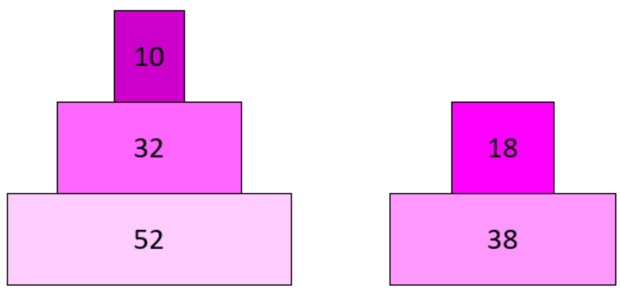
2. (15 + 20 points) **Greedy Algorithm.** Jane is a baker and she bakes cakes of varying sizes for her customers. She has to pack the cakes in boxes and send them out for delivery at the end of the day. She bakes the cakes in order of when the customer places their order, and she must pack the cake before starting the next cake. She can use as many boxes as she needs, but the delivery cost is dependent on the number of boxes she uses. She can stack several cakes in the same box, but she cannot stack a larger cake onto a smaller cake or it will collapse. Our objective is to help Jane minimize her deliverly cost, in other words the number of boxes she needs to use.

More precisely the input to our algorithm is an ordered list of cakes' sizes e.g., [52, 32, 38, 10, 18]. The output is a list of valid stacks of cakes. A valid stack should be non-collapsing e.g., the stack $(32, 38)$ would collapse since the larger cake (size 38) is on top of the smaller one (size 32). Similarly, a valid stack should respect the original ordering of cakes e.g., the stack $(38, 32)$ is invalid because the cake with size 38 would have been baked/packed after the cake with size 32. The goal is to minimize the number of stacks of cakes. For this problem we will assume that there is no limit on the height of a stack.

Since cakes must be stacked in order, we could, for example, stack cake 1 on box 1, cake 2 on box 2, cake 3 on box 3, cake 4 on box 2, and cake 5 on box 1 as shown below.



However, the optimal solution is 2 boxes as follows.



(a) Devise a greedy algorithm which returns a packing of the cakes that minimizes her delivery cost. Analyze the time and space complexity of your algorithm.

(b) Prove the correctness of your algorithm i.e. prove that the number of boxes used by your greedy algorithm is no more than number of boxes used by any optimal algorithm.

3. (30 points) Consider a *directed* graph $G : (V, E)$ with non negative edge weights in which each edge is also assigned a color, either white, black, or pink. Given $s, t \in V$, our objective is to find the length of the *shortest walk* (least cost) from s to t such that edges in the path have colors repeated in the following order: (white, black, pink). For instance, a s-t walk in which edges have the colors (white, black, pink, white, black) or (black, pink, white, black, pink, white) is valid. Note that a walk (as opposed to a path) allows repetition of vertices.

   Hint: Create a new graph $G' : (V', E')$ such that V' includes three copies of each vertex in V and $|E'| = |E|$. Think about how to add edge $(u, v) \in E$ to E' based on its color.

4. **(*) 1st Bonus Problem.** (10 + 20 points)—*Recall that this problem will be graded separately, and the grade will not count towards the grade of this homework. The solution to this problem, along with two more bonus problems to be included in later homeworks, will count towards an extra 10% on the final grade of the class.*

(a) Let T_1 and T_2 be two spanning trees, and let $e \in T_1 \setminus T_2$ be an edge in T_1 but not T_2. Prove that there exists an edge $d \in T_2 \setminus T_1$ such that $T_2 - d + e$ and $T_1 - e + d$ are both spanning trees. (Here "+" denotes adding an edge and "-" denotes deleting an edge.)

(b) Let T_1 and T_2 be two spanning trees. Show that there exists a one-to-one mapping $f : (T_2 \setminus T_1) \to (T_1 \setminus T_2)$ such that for every $e \in T_2 \setminus T_1$, the three $T_1 - f(e) + e$ is a spanning tree.

5. **2 points** Have you assigned pages on Gradescope?