Problem 1

(a) Design a data structure to support the following two operations for a dynamic multiset S of integers, which allows duplicate values:

1. INSERT(S, x) inserts x into S
2. DELETE-LARGER-HALF(S) deletes the largest $\lceil |S|/2 \rceil$ elements from S

Show how to implement this data structure so that any sequence of m INSERT and DELETE-LARGER-HALF operations runs in $O(m)$ time. Your implementation should also include a way to output the elements of S in $O(|S|)$ time. You can assume that your initial multiset is empty.

Hint: to delete larger half, need to find the median. You can use the result that there is a linear time algorithm for finding the median in an unordered array (if you don't know this result, you should learn about it - it is pretty cool).

(b) An **ordered stack** is a data structure that stores a sequence of items and supports the following operations:

1. ORDEREDPUSH(x) removes all items smaller than x from the beginning of the sequence and then adds x to the beginning of the sequence.
2. POP() deletes and returns the first item in the sequence, or NULL if the sequence is empty.

Suppose we implement an ordered stack with a simple linked list, using the obvious ORDEREDPUSH and POP algorithms. Prove that if we start with an empty data structure, the amortized cost of each ORDEREDPUSH or POP operation is $O(1)$.

Problem 2

In this problem we will work out some splaying counterexamples;

(a) Single rotations "don't work" for splay trees. To show this, consider a degenerate n-node "linked-list shaped" binary tree where each node's right child is empty. Suppose that the (only) leaf is splayed to the root by *single* rotations. Show the structure of the tree after this splay. Generalize this to argue that there is a sequence of $n/2$ splays (each splay using only single rotations) that each take at least $n/2$ work.

(b) Now from the same starting tree, show the final structure after splaying the leaf with (zig-zig) double rotations. Explain how this splay has made much more progress than single rotations in "improving" the tree.

(c) Given the theorem about access time in splay trees, it is tempting to conjecture that splaying does not create trees in which it would take a long time to find an item. Show this conjecture is false by showing that for large enough n, it is possible to restructure any binary tree on n nodes into any other binary tree on n nodes by a sequence of splay requests. Conclude that it is possible to make a sequence of requests that cause the splay tree to achieve any desired shape.

Hint: start by showing how you can use splay requests to make a specified node into a leaf, then recurse.

Problem 3

The simplest model for a random graph consists of n vertices, and tossing a fair coin for each pair $\{i, j\})$ to decide whether this edge should be present in the graph. Call this model $G(n, 1/2)$. A triangle is a set of 3 vertices with an edge between each pair.

1. What is the expected number of triangles?
2. What is the variance?
3. Show that the number of triangles is concentrated around the expectation and give an expression for the bound in the decay of probability.

Problem 4

In this problem, we are in the setting where given a set S (which is not known to us - and this set does not have repeated elements), we only have access to S by querying a random element from S uniformly at random. Thus, if we want to sample s elements from S, we will obtain a sequence of elements $a_1, \ldots, a_s \in S$ where each a_k was drawn from S uniformly at random. Thus it could be the case where $a_i = a_j$ for some $i \neq j$.

1. Show that given n distinct numbers in $[0, 1]$ it is impossible to estimate the value of the median within say 1.1 multiplicative approximation factor with $o(n)$ samples.

   Hint: to show an impossibility result you show two different sets of n numbers that have very different medians but which generate, with high probability, identical samples of size $o(n)$.

2. Now calculate the number of samples needed (as a function of t) so that the following is true: with high probability, the median of the sample has at least $n/2 - t$ numbers (from the given set of n numbers) less than it and at least $n/2 - t$ numbers (from the given set of n numbers) more than it.

Problem 5

Consider again the experiment in which we toss m labeled balls at random into n labeled bins, and let the random variable X be the number of empty bins. We have seen that $\mathbb{E}[X] = n \cdot \left(1 - \frac{1}{n}\right)^m$.

(a) By writing $X = \sum_i X_i$ for suitable random variables X_i, show how to derive the following exact formula for the variance of X:

$$Var[X] = n \cdot \left(1 - \frac{1}{n}\right)^m + n(n-1) \cdot \left(1 - \frac{2}{n}\right)^m - n^2 \cdot \left(1 - \frac{1}{n}\right)^{2m}$$

(b) What is the asymptotic value (as $n \to \infty$) of $Var[X]$ in the cases $m = n$ and $m = n \ln(n)$?

Hint: you may use the approximations $(1-x/n)^n \sim e^{-x} \cdot (1-x^2/2n)$ and $(1-1/n)^{xn} \sim e^{-x} \cdot (1-x/2n)$

(c) When throwing n balls into n bins, what is the expected number of bins with *exactly* one ball? Compute an exact formula and its asymptotic approximation.

Problem 6

1. Let us consider the coupon collector problem: we toss $m = n \log n + cn$ balls into n bins, where c is a constant, and we are interested in the probability that there is no empty bin. We saw in class that

$$Pr[\text{some bin is empty}] \leq n \cdot \left(1 - \frac{1}{n}\right)^m \sim n \cdot \frac{1}{e^{m/n}} = \frac{1}{e^c}.$$

   Prove that

$$Pr[\text{some bin is empty}] = \Omega\left(\frac{1}{e^c} - \frac{1}{2e^{2c}}\right).$$

   Hint: inclusion-exclusion

2. Consider the following variation of the coupon collector's problem. Each box of cereal contains one of $2n$ different coupons. The coupons are organized into n pairs, so that coupons 1 and 2 are a pair, coupons 3 and 4 are a pair, and so on. Once you obtain one coupon from every pair, you can obtain a prize. Assuming that each coupon in each box is chosen independently and uniformly at random from the $2n$ possibilities, what is the expected number of boxes you must buy before you can claim the prize?