Homework may be completed in group of size with at most three students. You’re not allowed to use material from the web or talk about the homework with anybody outside your collaboration group.

- Solutions should be typed and uploaded as a pdf to gradescope.com (instructions to follow).
- To get full marks, answers must be sufficiently detailed, supported with rigorous proofs (for correctness and running time). Faster algorithms will typically get more marks than slower algorithms.

**Question 1.** We want to design an algorithm that, given a list of distinct integers \( y_1, y_2, \ldots, y_n \) distinguishes between Case 1: The list is sorted, i.e., \( y_1 < y_2 < \ldots < y_n \) and Case 2: The list is \( \epsilon \)-far from being sorted, i.e., greater than \( \epsilon n \) elements need to be changed to make the list sorted. If the list is sorted, the algorithm should accept the input with probability 1 and if the list is \( \epsilon \)-far from being sorted the algorithm should reject with probability at least \( \frac{3}{4} \).

1. Consider the algorithm that randomly samples \( i \in [n-1] \) and checks whether \( y_i < y_{i+1} \). Construct an input that is \( \frac{1}{2} \)-far from being sorted for which this approach would require \( \Omega(n) \) queries to determine the list isn’t sorted.
2. Consider the algorithm that randomly samples \( i, j \in [n] \) and checks whether \( y_i < y_j \). Construct an input that is \( \frac{3}{4} \)-far from being sorted for which this approach would require \( \Omega(n) \) queries to determine the list isn’t sorted.
3. Consider the algorithm that random samples \( i \in [n] \); queries \( y_i \) and sets \( z \) to be this value; and then performs a binary search in the list for the value \( z \). If the values of the list queried in the search for \( z \) reveal that the list isn’t sorted, reject the output. Prove that repeating this process \( O(1/\epsilon) \) times and accepting the input if none of the binary searches reject the input yields an efficient tester. **Hint:** Call index \( i \) good if the search for \( y_i \) doesn’t lead to rejection and show that the values at good indices are sorted.

**Question 2.** Given a set \( X \) of points in any metric space, assume that you can compute the distance between any pair of points with one query. Say that \( X \) is \((k,b)\)-diameter clusterable if \( X \) can be partitioned into \( k \) subsets (clusters) such that the maximum distance between any pair of points in a cluster is \( b \). Say that \( X \) is \( \epsilon \)-far from \((k,b)\)-diameter clusterable if at least \( \epsilon |X| \) points must be deleted from \( X \) in order to make it \((k,b)\)-diameter clusterable. Show how to distinguish the case when \( X \) is \((k,b)\)-diameter clusterable from the case when \( X \) is \( \epsilon \)-far from \((k,2b)\)-diameter clusterable. Your algorithm should use polynomial in \( k, 1/\epsilon \) queries.

**Question 3.** Two rooted trees \( T_1 \) and \( T_2 \) are said to be *isomorphic* if there exists a bijection from the vertices of \( T_1 \) to the vertices of \( T_2 \) satisfying the following condition: for each internal vertex \( v \) of \( T_1 \), if \( v_1, v_2, \ldots, v_k \) are children of \( v \) in \( T_1 \) then \( f(v_1), f(v_2), \ldots, f(v_k) \) are the children of \( f(v) \) in \( T_2 \). Note that no ordering is assumed on the children of any internal vertex.

1. Given a rooted tree \( T \), associate a multivariate polynomial \( P_v(x_0, x_1, x_2, \ldots) \) with each vertex \( v \) in the tree as follows: If \( v \) is a leaf let \( P_v = x_0 \) and if \( v \) has height \( h > 0 \) (i.e., the maximum distance from \( v \) to a descendant leaf is \( h \)) then let \( P_v \)

\[
(x_h - P_{v_1})(x_h - P_{v_2}) \ldots (x_h - P_{v_k}) .
\]
Prove that for any \( v \in T_1, w \in T_2 \), the subtree of \( T_1 \) rooted at \( v \) is isomorphic to the subtree of \( T_2 \) rooted at \( w \) iff \( P_v = P_w \).

(2) Design and analyze an efficient randomized algorithm for testing whether \( T_1 \) and \( T_2 \) are isomorphic. **Hint:** You may wish to appeal to the Schwartz-Zippel Lemma (proved in COMPSCI 611) that states that if \( P_1(x_0, x_1, \ldots) \) and \( P_2(x_0, x_1, \ldots) \) are different multivariate polynomial of total degree \( d \) and \( S \) is a fixed set of values then,

\[
P \left[ P_1(r_0, r_1, \ldots) = P_2(r_0, r_1, \ldots) \right] \leq d/|S|
\]

where \( r_0, r_1, \ldots \) are chosen independently and uniformly from the set \( S \).

**Question 4.** Let \( p \) be a prime number and let \( a, b \) be chosen uniformly at random from \( \{0, 1, \ldots, p-1\} \). Define \( p \) random variables \( Y_0, \ldots, Y_{p-1} \) where \( Y_i = ai + b \mod p \). Prove that these random variables are pairwise independent, i.e., for \( i \neq j \) and all \( \alpha, \beta \in \{0, 1, \ldots, p-1\} \):

\[
P \left[ Y_i = \alpha, Y_j = \beta \right] = P \left[ Y_i = \alpha \right] P \left[ Y_j = \beta \right].
\]

Prove or disprove that the random variables 3-wise independent, i.e., for distinct \( i, j, k \) and all \( \alpha, \beta, \gamma \in \{0, 1, \ldots, p\} \), is it true that \( P \left[ Y_i = \alpha, Y_j = \beta, Y_k = \gamma \right] = P \left[ Y_i = \alpha \right] P \left[ Y_j = \beta \right] P \left[ Y_k = \gamma \right] \)?

Let \( f \) be some function such that for all \( i \), \( f(Y_i) = 1 \) with probability \( q \) and \( f(Y_i) = 0 \) with probability \( 1-q \). Let \( X = \sum_i f(Y_i) \). Prove the best upper bound you can on \( P [X = 0] \). **Hint:** Think Chebyshev.

**Question 5.** Consider a 2-wise random hash function \( h : [n] \to [w] \), i.e., each \( h(i) \) is distributed uniformly at random from \( [w] \) and \( h(i) \) and \( h(j) \) are independent if \( i \neq j \). Let \( c \) be a 4-wise random function \( c : [n] \to \{-1, 1\} \). Let \( f = (f_1, f_2, \ldots, f_n) \in \mathbb{R}^n \) and \( X = \sum_{i \in [n]} (\sum_{j \in [n]} h(j) = i f_j c(j))^2 \).

Prove that,

1. \( E [X] = \sum_{i \in [n]} f_i^2 =: F_2 \)
2. \( \forall [X] = O(F_2^2 / w) \)

Let \( a_1, \ldots, a_m \) be a stream where each \( a_i \in [n] \) and define \( f_i = |\{ j : a_j = i \}| \). Design a small space data stream algorithm that approximates \( F_2 \) such that the estimate \( \hat{F}_2 \) satisfies

\[
P \left[ |\hat{F}_2 - F_2| \leq \epsilon F_2 \right] \geq 1 - \delta.
\]