PROBLEM 1 (20 Points) - Fingeprinting

Consider the problem of deciding whether two integer *multisets* S_1 and S_2 are identical (that is, each integer occurs the same number of times in both sets). This problem can be solved by sorting the two sets in $O(n \log n)$ time, where $n = |S_1| = |S_2|$. In this question, you will devise 2 faster randomized algorithms for this problem.

You can assume that the multisets S_i only have integers of bit complexity w, and that integer operations of $O(w)$-bit integers can be executed in $O(1)$ time (RAM model), and that a prime with $O(w)$-bits can be found in $O(n)$ time.

1. Use polynomial identity testing to give a $O(n)$ time algorithm for the problem above.

2. Use hashing to give an algorithm which runs in expected $O(n)$ time for the problem above.

Your algorithm for both parts should succeed with probability $\geq 2/3$.

PROBLEM 2 (10 Points) - Randomized Matching

Given a randomized algorithm $\mathcal{A}$ for testing the existence of a perfect matching in a graph G, give a randomized algorithm for finding a perfect matching in G. Analyze the running time and success probability of your algorithm on graphs with n vertices and m edges, given that the algorithm $\mathcal{A}$ runs in time $T(n, m)$, and has failure probability $\leq 1/n$.

Your algorithm should have success probability $\geq 2/3$, and it should definitely be guaranteed to stop after $\mathsf{poly}(n, m)T(n, m)$ time.

PROBLEM 3 (15 Points) - Sublinear Time Algorithms

Given a graph G of max degree d (*as adjacency list*), and a parameter $\epsilon > 0$, give an algorithm which has the following behavior: if G is connected, then the algorithm should pass with probability 1, and if G is ϵ-far from connected (at least $\epsilon \cdot n \cdot d$ edges must be added to connect G), then the algorithm should fail with probability at least 3/4. Your algorithm should look at a number of edges that is independent of n, and polynomial in d, ϵ.

For this problem, when proving the correctness of your algorithm, it is ok to show that if the input graph G is likely to be passed, then it is ϵ-close to a graph G_0 which is connected, without requiring that G_0 has degree at most d.

Problem 4 (15 Points) - Markov Chains

1. Show that if a Markov chain with transition matrix P is irreducible and has a state i such that $P_{i,i} > 0$, then it is also aperiodic.

2. Let a, b be positive integers and consider the Markov chain with state space

$$\{(i,j) \mid 0 \leq i \leq a-1, \quad 0 \leq j \leq b-1\}$$

where i, j are integers, and the following transition mechanism: if the chain is in state (i,j) at time t, then at time $t+1$ it moves to $((i+1) \bmod a, j)$ with probability $1/2$ or to $(i, (j+1) \bmod b)$ with probability $1/2$.

Show that this Markov chain is irreducible, and show that it is aperiodic if, and only if, $\gcd(a,b) = 1$.

3. Consider a chessboard with a lone white king making random (king) moves, meaning that at each move, he picks one of the possible squares to move to, uniformly at random. Is the corresponding Markov chain irreducible and/or aperiodic? If so, what is the stationary distribution?

Problem 5 (10 Points) - Random Walks

1. Consider a Markov chain on the vertices of a triangle: the chain moves from a vertex to one of the other two vertices with probability $1/2$ each. Find the stationary distribution and ε-mixing time of this Markov chain. Also, find the probability that the chain is at the starting vertex after n steps, for $n \geq 1$.

2. Suppose that we alter the transition probabilities of the Markov chain in the previous question as follows:

$$p_{12} = p_{23} = p_{31} = 2/3, \quad p_{21} = p_{32} = p_{13} = 1/3.$$

Find the stationary distribution and the ε-mixing time of this Markov chain. Also, find the probability that the chain is at the starting vertex after n steps, for $n \geq 1$.

Problem 6 (15 Points) - Linear Programming

Let $S = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ and $T = \{x \in \mathbb{R}^n \mid Bx \leq c\}$, where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $B \in \mathbb{R}^{t \times n}$ and $c \in \mathbb{R}^t$. Given A, B, b, c as an input, give a polynomial time algorithm for the problem of checking whether $S \subset T$.

Problem 7 (15 Points) - LP duality

For a directed graph $G(V, E)$, with 2 special vertices $s, t \in V$, let c_e be the capacity of edge $e \in E$ and consider the max-flox LP:

$$\max_{P} \sum_{s\text{-}t \text{ path of } G} f_P$$

subject to

$$\sum_{e \in P} f_P \leq c_e \quad \forall e \in E,$$
$$0 \leq f_P \quad \text{for all paths } P.$$

The first constraint says that the total flow on all paths from s to t is at most the capacity of each edge, and the second constraint says that the flow on each path is non-negative.

Give the dual of this LP and explain how to interpret the dual LP as a minimum cut problem.