CSE 632: Analysis of Algorithms II: Randomized Algorithms — Fall 2019

Homework 2

Lecturer: Shi Li — **Deadline: Oct 15, 2019**

Problem 1 (15 points) Let $p \geq 2$ be a prime. For every $a, b \in \{0, 1, 2, \cdots, p-1\}$, we define a function $h_{a,b} : \{0, 1, 2, \cdots, p-1\} \times \{0, 1, 2, \cdots, p-1\} \to \{0, 1, 2, \cdots, p-1\}$ as

$$h_{a,b}(x, y) = ax + by \bmod p, \quad \forall x, y \in \{0, 1, 2, \cdots, p-1\}.$$

Let $\mathcal{H}$ be the uniform distribution over all functions $h_{a,b}, a, b = 0, 1, 2, \cdots, p-1$. Prove that $\mathcal{H}$ is a universal hash distribution.

Problem 2 (30 points) Consider the deterministic quick-sort algorithm where we always choose the first element in A as the pivot. Also, assume we construct the array B (C) in such a way that it contains the elements in A smaller (bigger) than the pivot, in the same order as they appear in A. For simplicity, we assume all the elements in the input array are distinct.

(2a) (5 points) Give an array A of size n where the algorithm runs in time $\Omega(n^2)$.

(2b) (10 points) Show how to produce an uniformly random permutation of A in $O(n)$ time. That is, the probability that we output each permutation of A is exactly $1/n!$ (notice that there are $n!$ different permutations of A).

(2c) (15 points) Now consider the following modification of the quick-sort algorithm : given an input array A, we first produce an uniformly random permutation A' of A, and then run the above deterministic quick-sort algorithm on the permuted array A'. Show that this new algorithm has expected running time $O(n \log n)$.

Problem 3 (15 points) Let $G = (V, E)$ be a connected graph with $|V| = n$ and s be the size of the global minimum cut of G. Using the analysis of the Karger's algorithm to show that there are at most $O(n^2)$ different minimum cuts. (Notice that the $O(n^2)$ bound is tight: Consider a cycle of n vertices, which has global minimum cut size $s = 2$. Then any 2 edges on the cycle form a minimum cut and thus there are $n(n-1)/2$ minimum cuts.)

As a bonus problem, you can try to prove that for any integer $t \geq 1$, the number of different cuts in G of size at most ts is at most $O(n^{2t})$.