Algorithms, Summer 2019 at CIS

Homework 1 **Due: 7/18/19 before class**

---

1. Suppose you are in the comparison-based model and you are given a list of n distinct numbers, $a_1, a_2, a_3, \ldots, a_n$. Define the *rank* of an item a_j to be its position i, in a sorted order of these numbers. So if π is a permutation from $\{1, 2, \ldots, n\}$ to $\{1, 2, \ldots, n\}$, and $a_{\pi(1)} < a_{\pi(2)} < \cdots < a_{\pi(n)}$, then the rank of a_j is equal to the index i for which $\pi(i) = j$.

   Give a deterministic algorithm for outputting the entire set of items of rank 3^i, for $i = 0, 1, 2, 3, 4, \ldots, \lfloor \log_3 n \rfloor$. Your algorithm should use $O(n)$ comparisons.

2. Suppose you are in the comparison-based model are you are given a list of n distinct numbers, $a_1, a_2, a_3, \ldots, a_n$. You are also given an integer B, and suppose B divides n. Your job is to arbitrarily partition these n numbers into B groups $G_1, \ldots, G_B$, so that

   (a) each group G_i has n/B items, and

   (b) inside of each group G_i, the numbers are sorted.

   First argue that if $B = \Theta(n)$, then this can be done deterministically using $O(n)$ comparisons. Second, show that if $B = \Theta(n^{1/3})$, then this requires $\Omega(n \log n)$ comparisons for any deterministic algorithm in the comparison-based model.

3. Suppose there are m distinct integers $a_1, \ldots, a_m$ which are each drawn from the universe $U = \{0, 1, 2, 3, ..., n-1\}$. We would like to choose a random hash function h from a family H so that for all $i \neq j$, $h(a_i) \neq h(a_j)$, that is, the set $\{a_1, \ldots, a_m\}$ is perfectly hashed under h.

   In lecture we will show that if H is universal and has range size $M = \Theta(m^2)$, then a random h in H has this property with constant probability. However, for each of the hash functions we saw, if $a_1, \ldots, a_m$ are not fixed in advance, then specifying a random h in H requires at least $\log n$ bits. We would like to use fewer random bits when m is much less than n.

   Suppose we choose a random prime p among the first $10m^2 \cdot \log_2 n$ primes. Consider the map $h : U \to \{0, 1, 2, ..., 10m^2 - 1\}$ given by $h(y) = g(y \bmod p)$, where g is a universal hash function with range size $10m^2$. Argue that h has the above perfect hashing property with probability at least $9/10$. How many bits are needed to specify h? You may use the fact that the i-th prime number $p_i = \Theta(i \log i)$.