CS648A : Randomized Algorithms

Department of Computer Science and Engineering, IIT Kanpur

1 Randomized quick select

Let S be a set of n real numbers. Consider the randomized algorithm *Rand-QSelect*(k, S) described below that finds the k^{th} smallest element from the set S.

Select a pivot element x uniformly randomly from set S. Find its rank in the set S (by comparing x with every other element of set S). Let r be the rank of x. If $r = k$, we report x as the output. Otherwise we proceed recursively as follows: If $r > k$, then *Rand-QSelect*$(k, S_{<x})$ Else *Rand-QSelect*$(k - r, S_{>x})$.

Where $S_{<x}$ and $S_{>x}$ are the sets consisting of all those elements that are respectively smaller and greater than the element x. Observe that the running time of the above algorithm is dominated by the number of comparisons performed. Therefore, in order to get a bound on the expected running time of the algorithm, our aim is essentially to find out the expected number of comparisons performed in *Rand-QSelect*(k, S). Prove the following statements.

1. The expected number of comparisons is at most $3.5n$.
2. There are elements in set S which will be compared expected $\Theta(\log n)$ times during the algorithm.

2 Making an intelligent guess

We have a function $F : \{O, ..., n-1\} \to \{O, ..., m-1\}$. We know that, for $0 \leq x, y \leq n-1$,

$$F((x + y) \bmod n) = (F(x) + F(y)) \bmod m$$

The only way we have for evaluating F is to use a lookup table that stores the values of F. Unfortunately, an Evil Adversary has changed the value of $1/5$ of the table entries when we were not looking. Describe a simple randomized algorithm that given an input z, outputs a value that equals $F(z)$ with probability at least $1/2$. Your algorithm should work for every value of z, regardless of what values the Adversary changed. Your algorithm should use as few lookups and as little computation as possible.

Suppose I allow you to repeat your initial algorithm k times. What should you do in this case, and what is the probability that your *enhanced* algorithm returns the correct answer?