CS5314 Randomized Algorithms

Assignment 2
Due: 2:10 pm, Nov 3, 2011 (before class)

1 Questions

1. Generalize the median-finding algorithm to find the kth largest item in a set of n items for any given value of k. Prove that your resulting algorithm is correct, and bound its running time.

2. A fixed point of a permutation $\pi : [1, n] \to [1, n]$ is a value for which $\pi(x) = x$. Find the variance in the number of fixed points of a permutation chosen uniformly at random from all permutations.

   Hint: Let X_i be an indicator such that $X_i = 1$ if $\pi(i) = i$. Then, $\sum_{i=1}^{n} X_i$ is the number of fixed points. You cannot use linearity to find $\mathbf{Var}[\sum_{i=1}^{n} X_i]$, but you can calculate it directly.

3. The weak law of large numbers state that, if $X_1, X_2, X_3, \ldots$ are independent and identically distributed random variables with finite mean μ and finite standard deviation σ, then for any constant $\varepsilon > 0$ we have

   $$\lim_{n\to\infty} \Pr\left(\left|\frac{X_1 + X_2 + \cdots + X_n}{n} - \mu\right| > \varepsilon\right) = 0.$$

   Use Chebyshev's inequality to prove the weak law of large numbers.

4. If X is a binomial random variable $Bin(n, 1/2)$ with $n \geq 1$, show that the probability that X is even is $1/2$.

5. Prove that $\mathrm{E}[X^k] \geq (\mathrm{E}[X])^k$ for any positive even integer k.

6. (Further studies: No marks) For a random variable X with standard deviation $\sigma(X)$, and any positive real number t, Chebyshev's inequality implies that

   $$\Pr(X - \mathrm{E}[X] \geq t\sigma(X)) \leq \Pr(|X - \mathrm{E}[X]| \geq t\sigma(X)) \leq \frac{1}{t^2}.$$

   Indeed, we can bound $\Pr(X - \mathrm{E}[X] \geq t\sigma(X))$ slightly tighter. Show that

   $$\Pr(X - \mathrm{E}[X] \geq t\sigma(X)) \leq \frac{1}{1 + t^2}.$$

7. (Further studies: No marks) Previously, we have shown the expected time for the Randomized Quicksort to sort n numbers is $O(n \log n)$. Here, we want to prove further that the algorithm runs in $O(n \log n)$ time with high probability. Consider the following view of Randomized Quicksort. Every point in the algorithm where it decides on a pivot element is called a *node*. Suppose the size of the set to be sorted at a particular node is s. The node is called *good* if the pivot element divides the set into two parts, each of size not exceeding $2s/3$. Otherwise, the node is called *bad*.

   The nodes can be thought of as forming a tree in which the root node has the whole set to be sorted and its children have the two sets formed after the first pivot step and so on.

(a) Show that the number of good nodes in any path from the root to a leaf in this tree is not greater than $c \log_2 n$, where c is some positive constant.

(b) Show that, with high probability (greater than $1 - 1/n^2$), the number of nodes in a given root to a leaf path of the tree is not greater than $c' \log_2 n$, where c' is another constant. (*Hint:* If c' is much larger than c, there must be many bad nodes on the path... Will that be likely to occur?)

(c) Show that, with high probability (greater than $1 - 1/n$), the number of nodes in the *longest* root to leaf path is not greater than $c' \log_2 n$. (*Hint:* How many nodes are there in the tree?)

(d) Use your answers to show that the running time of Randomized Quicksort is $O(n \log n)$ with probability at least $1 - 1/n$.