CS 473: Fundamental Algorithms, Spring 2011

HW 7

Question 1 is due by **Sunday, 23:59:59, March 27** Questions 2-4 are due by **Monday, 23:59:59, March 28**

This homework contains four problems. **Read the instructions for submitting homework on the course webpage**.

Collaboration Policy: For this homework, Problems 2–4 can be worked in groups of up to three students.

Problem 1 should be answered in Compass as part of the assessment HW7-Online and should be done individually.

---

1. HW7-Online. (20 pts.)

2. Voting tree (30 pts.)

   Consider a uniform rooted tree of height h (every leaf is at distance h from the root). The root, as well as any internal node, has 3 children. Each leaf has a boolean value associated with it. Each internal node returns the value returned by two or three of its children. The evaluation problem consists of determining the value of the root; at each step, an algorithm can choose one leaf whose value it wishes to read.

   (A) (30 pts.) Consider the recursive randomized algorithm that evaluates two subtrees of the root chosen at random. If the values returned disagree, it proceeds to evaluate the third sub-tree. If they agree, it returns the value they agree on. Show the expected number of leaves read by the algorithm on any instance is at most $n^{0.9}$.

   (B) (Extra credit.) (10 pts.) Show that for any deterministic algorithm, there is an instance (a set of boolean values for the leaves) that forces it to read all $n = 3^h$ leaves. (hint: Consider an adversary argument, where you provide the algorithm with the minimal amount of information as it request bits from you. In particular, one can devise such an adversary algorithm.).

3. Conditional probabilities and expectations. (30 pts.)

   Assume there are two random variable X and Y, and you know the value of Y (say it is y). The ***conditional probability*** of X given Y, written as $\mathbf{Pr}\left[X \,\middle|\, Y\right]$, is the probability of X getting the value x, given that you know that $Y = y$. Formally, it is

$$\mathbf{Pr}\left[X = x \,\middle|\, Y = y\right] = \frac{\mathbf{Pr}[X = x \cap Y = y]}{\mathbf{Pr}[Y = y]}.$$

   The ***conditional expectation*** of X given Y, written as $\mathbf{E}\left[X \,\middle|\, Y = y\right]$ is the expected value of X if you know that $Y = y$. Formally, it is the function

$$f(y) = \mathbf{E}\Big[X \;\Big|\; Y = y\Big] = \sum_{x \in \Omega} x \,\mathbf{Pr}\Big[X = x \;\Big|\; Y = y\Big].$$

(A) (2 pts.) Prove that if X and Y are independent then $\mathbf{Pr}\Big[X = x \;\Big|\; Y = y\Big] = \mathbf{Pr}[X = x]$.

(B) (2 pts.) Let X_i be the number of elements in **QuickSelect** in the ith recursive call, when starting with $X_0 = n$ elements. Prove that $\mathbf{E}\Big[X_i \;\Big|\; X_{i-1}\Big] \leq (3/4)X_{i-1}$.

(C) (2 pts.) Prove that for any discrete random variables X and Y it holds $\mathbf{E}[\mathbf{E}[X|Y]] = \mathbf{E}[X]$.

(D) (10 pts.) Prove that, in expectation, the ith recursive call made by **QuickSelect** has at most $(3/4)^i n$ elements in the sub-array it is being called on.

(E) (4 pts.) Let X be a random variable that can take on only non-negative values. Assume that $\mathbf{E}[X] = \mu$, where $\mu > 0$ is a real number (for example, μ might be 0.01). Prove that $\mathbf{Pr}[X \geq 1] \leq \mu$.

(F) (10 pts.) Using (D) and (E) prove that with probability $\geq 1 - 1/n^{10}$ the depth of the recursion of **QuickSelect** when executed on an array with n elements is bounded by $M = c \lg n$, where c is some sufficiently larger constant (figure out the value of c for which your claim holds!).
(Hint: Consider the random variable which is the size of the subproblem that **QuickSelect** handles if it reaches the problem in depth M, and 0 if **QuickSelect** does not reach depth M in the recursion.)

4. QUICKY MEDIAN. (20 pts.)

Let $A_1, A_2, \ldots, A_k$ be k ***sorted*** arrays where the size of A_i is n_i. Let $n = \sum_i n_i$ be the total number of elements in all the arrays (assume they are distinct numbers). Describe a randomized algorithm to find the median of the numbers in the combined set of arrays in $O(k \log^2 n)$ expected time. *Hint:* Adapt the randomized selection algorithm and make use of the fact that the arrays are sorted.