# HW 6 (due Monday, at noon, March 16, 2015)

vers **OLD CS 473: Fundamental Algorithms, Spring 2015** Version: **1.03**

**You also have to do the quiz online (on moodle).**

**Collaboration Policy:** For this homework, Problems 1–3 can be worked in groups of up to three students.

**1.** (40 PTS.) Not so quick select.

(A) (20 PTS.) You are given (implicitly) a set of $n$ numbers $S$. Given two numbers $\alpha < \beta$, you have a procedure **algCount** $(\alpha, \beta)$ which, in $O(n^{1/2} \log n)$ time, returns the number of elements in the set $S\langle \alpha, \beta \rangle = \left\{ x \in S \,\middle|\, \alpha < x < \beta \right\}$. You can assume that you have two numbers $\pm\infty$ that are smaller and larger than any number in the set $S$. Similarly, **algSample**$(\alpha, \beta)$ returns (in the same time bound as above), a number selected uniformly at random from the set $S\langle \alpha, \beta \rangle$. Present an algorithm, such that given $k$, it computes the $k$th smallest number of $S$. What is the expected running time of your algorithm (faster, is better, naturally). (Hint: Try to think how to implement **Quickselect** in this case.)

Note, that you can not access the set $S$ (or its elements) directly – your only access to $S$ is via **algCount** and **algSample**.

(B) (20 PTS.) You are given a three dimensional array $A[i, j, k]$, where $i.j, k$ can range from 1 to $n$. Furthermore, assume that for any $i, j, k$ we have that

(i) $A[i, j, k] < A[i + 1, j, k]$,

(ii) $A[i, j, k] < A[i, j + 1, k]$, and

(iii) $A[i, j, k] < A[i, j, k + 1]$.

Given a number $t$ between 1 and $n^3$, describe an algorithm that outputs the $t$th smallest number in $A$. What is the expected running time of your algorithm? (For credit, the running time of the algorithm needs to be significantly faster than $O(n^3)$).

Hint: Use the algorithm of (A) (you would have to modify the running time analysis, naturally).

**2.** (30 PTS.) Conditional probabilities and expectations.

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

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

(A) (2 PTS.) Prove that if $X$ and $Y$ are independent then $\mathbf{Pr}\left[X = x \,\middle|\, Y = y\right] = \mathbf{Pr}[X = x]$.

(B) (2 PTS.) Let $X_i$ be the number of elements in **QuickSelect** in the $i$th recursive call, when starting with $X_0 = n$ elements. Prove that $\mathbf{E}\left[X_i \,\middle|\, X_{i-1}\right] \leq (7/8)X_{i-1}$.

(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 $i$th recursive call made by **QuickSelect** has at most $(7/8)^i n$ 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, $\mu$ 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 large 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.)

**3.** (30 pts.) Random elections.

Consider an undirected graph $G = (V, E)$ with positive weights on the edges, with $n$ vertices and $m$ edges. Think about $G$ as modeling a communication network. A start-up called *NileChoice*, wants to deploy servers on all the nodes in the network (initially, it has no servers). Whenever a server is installed in a node, it becomes the server for all the nodes in the network that do not have a closer sever than the new one. In particular, whenever a server is brought online, all the clients that it serves, should receive a message to update their server information to the new server.

(A) (10 pts.) Consider a random permutation $\pi$ of the vertices, and consider a specific vertex $v$ in the network. Prove, that in expectation, if servers are installed according to the ordering of $\pi$, then $v$ in expectation would get $O(\log n)$ update messages (hint, what is the probability that in the $i$th iteration $v$ would get a message?).

(B) (5 pts.) Prove that in expectation, using a random permutation, would require (overall) sending $O(n \log n)$ messages.

(C) (15 pts.) Given the permutation $\pi$, describe an algorithm that computes all the messages that need to be sent if we create servers according to the ordering of $\pi$. What is the expected running time of your algorithm? (For full credit, the running time should be near linear.)