Randomized Algorithms, WS 2002/03
Problem Set 1
Nov 8, 2001

Lecturers:
Privatdozent Dr. Peter Sanders, Building 46, R.308, sanders@mpi-sb.mpg.de
Dr. Ernst Althaus, Building 46, R.304, althaus@mpi-sb.mpg.de
Home page: `http://www.mpi-sb.mpg.de/~sanders/courses/randalg/index.html`

This homework is due on Friday, Nov. 15.

Problem 1-1 (von Neumann's coin trick)
Suppose you are given a coin for which the probability of HEADS, say p, is unknown. How can you use this coin to generate unbiased (i.e., $\mathbf{Pr}[HEADS] = \mathbf{Pr}[TAILS] = \frac{1}{2}$) coin flips? Give a scheme for which the expected number of flips of the biased coin for extracting one unbiased coin-flip is no more than $\frac{1}{p(1-p)}$.

Hint: Consider two consecutive flips of the biased coin.

Problem 1-2 (Quicksort)
Suppose the input S of the quicksort algorithm is stored in an array. A recursive call sorts an interval of the array. Consider a deterministic variant of quicksort that always selects the first element in the current interval of the array as the pivot element.
a) Which are the worst case instances of this algorithm and what is the running time on these instances.
b) Assume your input is a permutation of the numbers $\{1, \ldots n\}$, chosen uniformly at random. What is the average case running time of the above algorithm?

Problem 1-3 (k-Median) Consider the following algorithm to compute the kth smallest of a set of n elements.

Input: a set S of n different numbers and a number k between 1 and n

Output: the kth smallest element of S

Algorithm: Find(S, k)

1. Choose y form S uniformly at random
2. $S_1 := \{x \in S \mid x < y\}; S_2 := \{x \in S \mid x > y\}$
3. If $|S_1| \geq k$ then Find(S_1,k)
4. If $|S_1| = k - 1$ then Output y
5. If $|S_1| < k - 1$ then Find(S_2, $k - |S_1| - 1$)

Show that the expected running time of the algorithm is $O(n)$.

Hint: We think that it is *not* a good idea to use an approach similar to the analysis of Quicksort.