Institut für Informatik
Prof. Dr. Heiko Röglin
Dr. Melanie Schmidt

Randomized Algorithms and
Probabilistic Analysis
Summer 2016

Problem Set 2

Problem 1

Let `ALG` be a randomized algorithm with running time $\mathcal{O}(n^3 + n + \sqrt{n})$ that outputs an optimal solution (for an unspecified optimization problem) with probability at least $\frac{1}{\sqrt{n}\log n}$. Give a number ℓ of independent repetitions such that repeating `ALG` ℓ times and returning the best solution results in an algorithm with success probability at least $1 - \frac{1}{n^7}$. What is the running time of the resulting algorithm?

Problem 2

Prove that the `FastCut` algorithm (without repetitions) has a running time of $\mathcal{O}(n^2 \log n)$. Is this still true when t is set slightly larger, for example to $t := 1 + \lceil (3/4)n \rceil$?

Problem 3

We are given a data stream of numbers $a_1, a_2, a_3, \ldots$ (of unknown length) and want to sample one number s. However, instead of choosing every item in the stream with the same probability, we want to achieve the following: After seeing a_i, we want that

$$\Pr(s = a_j) = \begin{cases} 2^{-(i-1)} & \text{for } j = 1 \\ 2^{-(i-j+1)} & \text{for } j \in \{2, \ldots, i\} \end{cases}.$$

1. To make sure that this defines a discrete probability measure, show that the sum of the desired probabilities $\sum_{j=1}^{i} \Pr(a_j)$ after seeing a_i is always 1.
2. Adapt `ReservoirSampling` such that it stores a number s which is equal to the different elements in the data stream with the desired probabilities.

Problem 4

Consider the following recursive and randomized algorithm:

`RandomRecursion`(ℓ)

1. Print ℓ on the screen.
2. Toss a random coin.
3. If `Heads`, call `RandomRecursion`($\ell + 1$).
4. Toss a random coin.
5. If `Heads`, call `RandomRecursion`($\ell + 1$).

What is the probability that the call `RandomRecursion`(0) finishes running after a finite time (does not run forever)?