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

UNIVERSITÄT BONN

Randomized Algorithms and
Probabilistic Analysis
Summer 2020

Problem Set 3

Please hand in your solutions for this problem set via email (ipsarros@uni-bonn.de) until Monday May 11th.

Problem 1

1. A lively monkey types $26^6 \cdot 42 + 5$ letters (=499017796 letters) on a keyboard. We assume that the keyboard has only upper-case letters and that each of the 26 letters is chosen uniformly at random. What is the expected number of times that the word `RANDOM` appears?

2. We flip a fair coin $n + \log_2 n$ times, assume that n is a power of two. We get a sequence $x_1, x_2, \ldots, x_{n+\log_2 n}$ with $x_i \in \{H, T\}$. We say that $x_i, \ldots, x_{i+\ell-1}$ is an *ℓ-sequence* if $x_i = x_{i+1} = \ldots = x_{i+\ell-1}$ (all heads or all tails). What is the expected number of ℓ-sequences for $\ell = 1 + \log_2 n$?

Problem 2

As in the proof of Theorem 1.14 (success probability of Karger's Contract algorithm) let A_i be the event that the algorithm contracts a good edge in iteration i. Show or disprove that A_i and A_j are generally independent for $i, j \in \{1, \ldots, n-1\}$.

Problem 3

In this task, we want to cut a graph $G = (V, E)$ into r pieces instead of cutting it into two pieces as in the lecture. The parameter $r \in \mathbb{N}$ is a constant. We say that r disjoint subsets $V_1, \ldots, V_r$ with $V = \cup_{i=1}^{r} V_i$ are an *r-cut* of G. We pay for all edges between these subsets, our cost is: $\frac{1}{2}(|\delta(V_1)| + |\delta(V_2)| + \ldots + |\delta(V_r)|)$. We want to find an r-cut with minimum cost.

Generalize Karger's `Contract` algorithm such that it finds a minimum r-cut with probability $\Theta(1/n^{3r})$.

Problem 4

What is the running time of the `FastCut` algorithm (without repetitions) when we set $t := (3/4)n$? You may use the 'master theorem' (this theorem is explained in many books and lecture notes, see for example the notes from Avrim/Manuel Blum's course at `https://www.cs.cmu.edu/~avrim/451f11/lectures/lect0901.pdf`).

Master Theorem
The recurrence

$$T(n) = aT(n/b) + cn^k$$
$$T(1) = c,$$

where a, b, c and k are all constants, solves to:

$$T(n) \in \Theta(n^k) \text{ if } a < b^k$$
$$T(n) \in \Theta(n^k \log n) \text{ if } a = b^k$$
$$T(n) \in \Theta(n^{\log_b a}) \text{ if } a > b^k.$$