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

UNIVERSITÄT BONN

Randomized Algorithms and
Probabilistic Analysis
Summer 2020

Problem Set 6

Please hand in your solutions via e-mail (ipsarros@uni-bonn.de) until Monday June 1st.

Problem 1

Reconsider Lemma 3.17. Instead of using the maximum degree D to bound the running time of BFS in the explored subgraph by $D \cdot |V_i'|$, we now want to use $|V(G')|^2$ as a bound on $|E_i'|$. How does the outcome of the computation change, what running time bound do we get?

Problem 2

In the lecture, we saw that there is a permutation π for which the deterministic bit fixing strategy yields a routing with execution time $\Omega(2^n/n)$. Assume that instead of bit fixing paths as defined in the lecture, we use bit fixing paths where the bits are fixed in a random order. For each $i \in V_n$, the order is chosen uniformly at random from all permutations, and independently of the order chosen for all other $i' \in V_n$. Show that there is a permutation for which this routing has execution time $2^{\Omega(n)}$ with probability $1 - e^{-2^{\Omega(n)}}$. *Hint 1:* The same permutation as for the deterministic case works. *Hint 2:* You may use that Theorem 3.8 (2.) from the lecture still holds when $\mathbf{E}(X)$ is replaced by a lower bound $u \leq \mathbf{E}(X)$. *Hint 3:* You may use that it holds for all $i, j \in \mathbb{N}$, $i \geq j$, that

$$\left(\frac{i}{j}\right)^j \leq \binom{i}{j} \leq \left(\frac{i \cdot e}{j}\right)^j.$$

Problem 3

Assume that you have found a weird six-sided die which appears not to be a fair die, i.e. it seems that the probability for a roll of the die to land on a specific side is not necessarily the same for each side.

You want to estimate the probabilities for all sides of the die. More specifically, for each side you would like an estimate which is within 0.01 of the true probability with a certainty of at least 95%. How often do you have to roll the die to get estimates for all sides of the die which all satisfy this guarantee? Hint: Use Chebyshev's inequality.

Problem 4

The following is an example of *property testing in dense graphs*.

We say that a graph with n vertices is ε-far-away from being a complete graph if it is missing at least $\varepsilon \cdot n^2$ edges. Create an algorithm with sub-linear runtime that decides with high probability if a given graph is a complete graph or ε-far-away from being a complete graph.

In case the input is neither complete nor ε-far-away from being complete, the algorithm is allowed to return an arbitrary output. To interact with the graph you are only allowed to ask if a specific edge exists or not.

Recall that a graph $G = (V, E)$ is a complete graph if for every pair of vertices $u, v \in V$ with $u \neq v$, the graph contains the edge $\{u, v\}$.