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

UNIVERSITÄT BONN

Randomized Algorithms and
Probabilistic Analysis
Summer 2018

Problem Set 5

Problem 1

(Recall that $\sum_{i=1}^{n} \frac{1}{i} = H_n = \Theta(\log n)$)

1. Assume that we have n images, and that n is a multiple of k. Each image shows a portrait of a person, and there are k different persons. There are n/k images of each person. We want to have an (arbitrary) collection with exactly one picture of each person and use the following randomized algorithm: We keep choosing a picture uniformly at random (with replacement) until we have pictures of k different people. If we already have a picture of a person, we discard the chosen picture, otherwise we add it to our picture collection. What is the expected numer of times that we choose a picture until we have our collection of k pictures of different persons?

2. We want to sort n distinct numbers that are stored in array A. We use *GuessSort*: We pick two indices $i, j \in \{1, \dots, n\}$ uniformly at random from all possible pairs (i, j) with $i < j$. If $A[i] > A[j]$, we swap the elements, otherwise, we do nothing. Give an upper bound on the expected number of comparisons that this algorithm does until the array is sorted.

Problem 2

Which of the following deterministic algorithms compute a 2-approximation for the vertex cover problem? For each algorithm, give a counter example or argue why it provides a 2-approximation.

1. Start with the empty cover. As long as there are uncovered edges, pick an arbitrary uncovered edge. Add one arbitrary end point of the edge to the cover.

2. Start with the empty cover. As long as there are uncovered edges, pick an arbitrary uncovered edge. Add *both* end points of the edge to the cover.

3. Compute a maximal matching $M \subseteq E$. For every $e \in M$, add both end points to the cover. [A matching is a set of edges that contains no adjacent edges].

4. Compute a maximum cut $S \subset V$. Use S as the cover.

5. Run a depth first search and let T be the DFS tree of the search. Let I be the set of internal nodes of T, use I as the cover.

Problem 3

Can Markov's inequality be improved? Show the following statement. For any $a > 1$, it is possible to define a random variable X_a that is non-negative, $\mathbf{E}(X_a)$ exists, and it holds that

$$\mathbf{Pr}(X_a \geq a \cdot \mathbf{E}(X_a)) = 1/a.$$

Problem 4

Recall two concepts of randomized algorithms from the lecture.

- A Las-Vegas-Algorithm is a randomized algorithm with expected polynomial runtime which, once it terminates, always outputs the correct solution.
- A Monte-Carlo-Algorithm is a randomized algorithm with polynomial runtime which computes the correct solution with probability at least $1/2$. In case it did not compute the correct solution the algorithm can make any arbitrary output.

An example for a Las-Vegas-Algorithm is Randomized QuickSort from the lecture. An example for a Monte-Carlo-Algorithm can for example be obtained by $\Theta(\log n)$ repetitions of the FastCut algorithm.

- Given a Las-Vegas-Algorithm for a given problem, show how a Monte-Carlo-Algorithm for the same problem can be obtained.
- Given a Monte-Carlo-Algorithm for a given problem, under what extra assumption can we use the Monte-Carlo-Algorithm to obtain a Las-Vegas-Algorithm for the same problem?