Lehrstuhl für Informatik I
– Algorithmen und Komplexität –
Universität Würzburg

Würzburg, 13 December 2023

Johannes Zink
Oksana Firman

Exercise Sheet #6

Advanced Algorithms (WS 2023/24)

Exercise 1 – Packing argument

Let δ be a positive real number. Show that a $\delta \times \delta$ square can contain at most $\mathcal{O}(1)$ points such that the distance between any pair of these points is at least δ. **3 Points**

Exercise 2 – Randomized 3-coloring

Given a graph $G = (V, E)$ and the three colors red, green, and blue. Coloring the vertices, our goal is to maximize the number of bichromatic edges, that is, edges whose incident vertices have a different color. Design a simple Monte Carlo algorithm and analyze the expected number of bichromatic edges. **3 Points**

Exercise 3 – Fair coins and fair dice

We have a coin that shows tails with probability p and heads with probability $1 - p$. Suppose the coin is biased, that is, $p \neq 0.5$

a) Describe a strategy to get a fair result from the biased coin. **2 Points**

b) Can we use a similar strategy to make a biased k-sided die fair? Why or why not? What practical problem would we run into? **2 Points**

Exercise 4 – CautiousFindMin

Consider the following (unnecessarily slow) algorithm to compute the minimum of a set S of pairwise distinct integers.

If $|S| = 1$, return its unique element. Otherwise, pick a random element x and recursively compute the minimum x' of $S \setminus \{x\}$. If $x' < x$, return x'. Otherwise, the algorithm behaves overly cautious: check if x is smaller than all elements in $S \setminus \{x\}$ and if so, return x.

a) Is the algorithm correct and guaranteed to terminate? **1 Point**

b) What is the worst-case running time of the algorithm? **1 Point**

c) What is the expected running time of the algorithm? **3 Points**

Exercise 5 – Random permutation

Suppose you are given an array A of length n and you want to compute a random permutation of its elements. How can you do this with a fair coin (that can generate a random bit in $\mathcal{O}(1)$ time)? Describe a Las Vegas algorithm to find such a permutation. Show why it works correctly and show the expected running time. **5 Points**

Please hand in your solutions on Wuecampus until the beginning of the next lecture, that is 14:15 on Wednesday, December 20.