University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
S. Faour, M. Fuchs, A. Malyusz

Algorithm Theory
Exercise Sheet 11

Due: Friday, 17th of January 2025, 10:00 am

Exercise 1: Balls into Bins *(10 Points)*

Assume we have n bins and n balls (for $n \geq 2$). We now throw all the balls uniformly at random into the bins. In the following we want to show that the maximum number of balls per bin is at most $O(\log n)$ with high probability. For that we define the *maximum load* L by $\max_{1 \leq j \leq n} Y_j$ where (random variable) Y_j stands for the number of balls in bin j.

(a) For a given bin j, what is the expected number of balls in j? (i.e., compute $E[Y_j]$) *(2 Points)*

(b) Use a Chernoff Bound to show that $P(Y_j \geq 2e \cdot \log_2 n) \leq 1/n^{2e}$. *(6 Points)*
Chernoff Bound: Suppose $X_1, X_2, \ldots, X_N$ are *independent* random variables taking values in $\{0, 1\}$. Let X denote $\sum_{i=1}^{N} X_i$ and let $\mu = E[X]$ be this sums expected value. Then for any $\delta > 0$,

$$P\left(X \geq (1+\delta) \cdot \mu\right) \leq \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu}$$

(c) Show that the maximum load L is small, i.e., show that $P(L < 2e \cdot \log_2 n) > 1 - \frac{1}{n^4}$. Use a Union Bound! *(2 Points)*

Exercise 2: Max Cut *(10 Points)*

Let $G = (V, E)$ be a simple undirected graph. Consider the following randomized algorithm: Every node $v \in V$ joins set S with probability $1/2$. You can assume that $(S, V \setminus S)$ actually forms a cut i.e., $\emptyset \neq S \neq V$.

(a) Show that with probability at least $1/3$ this algorithm outputs a cut which is a 4-approximation to the maximum cut (i.e., the cut of maximum possible size) *(5 Points)*
Hint: Apply the Markov inequality to the number of edges that do ***not*** *cross the cut. For a non-negative random variable X, the Markov inequality states that for all $t > 0$ we have*

$$P(X \geq t) \leq \frac{E[X]}{t}$$

(b) How can you use the above's algorithm to devise a 4-approximation with probability at least $1 - \left(\frac{2}{3}\right)^k$ for any integer $k > 0$? *(4 Points)*

(c) How would you choose k from the previous subtask to make sure your algorithm computes a 4-approximation **with high probability**[1] ? *(1 Point)*

[1] We use the term **with high probability** in the context of graphs with n nodes and for *any* given constant $c > 0$ if the algorithms succeeds with probability at least $1 - \frac{1}{n^c}$.