University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
M. Ahmadi, P. Schneider

Algorithms Theory
Exercise Sheet 6

Due: Monday, 21st of January, 2019, 14:15 pm

Exercise 1: Contention Resolution *(2+2+4 Points)*

Consider the contention resolution problem explained in the lecture with n processes and a single shared resource. We would like to calculate the expected number of time slots until every process has been successful at least once. For all integers $i \leq n$, let random variable T_i denote the smallest integer such that exactly i different processes are successful to access the resource in the first T_i time slots.

(a) Let t be an arbitrary time slot in $[T_j + 1, T_{j+1}]$ for an arbitrary integer $j < n$. What is the probability that some process becomes successful for the first time in time slot t?

(b) For all $i \leq n$, let random variable X_i be the number of rounds needed for the i^{th} process to succeed after exactly $i-1$ distinct processes have succeeded, i.e., $X_i := T_i - T_{i-1}$. Then, for an arbitrary integer $j \leq n$, what is the expected value of X_j?

(c) What is the expected value of T_n, the time for all processes to succeed at least once?

Hint: The probability that some process is successful in a given time slot is $(1-1/n)^{n-1}$. *We have seen that this probability is approximately* $1/e$. *For simplicity, you can assume that this probability is exactly* $1/e$.

Exercise 2: Randomized Independent Set Algorithm *(6+7 Points)*

Let $G = (V, E)$ be a graph with n vertices and m edges. An independent set in a graph G is a subset $S \subseteq V$ of the nodes such that no two nodes in S are connected by an edge. Let $d := \frac{1}{n}\sum_{v \in V} \deg(v) = \frac{2m}{n}$ be the average node degree and consider the following randomized algorithm to compute an independent set S.

(I) Start with an empty set S. Then independently add each node of V with probability $1/d$ to S (you can assume that $d \geq 1$).

(II) The subgraph induced by S might still contain some edges and we therefore need to remove at least one of the nodes of each of the remaining edges. For this, we use the following deterministic strategy: As long as S is not an independent set, pick an arbitrary node $u \in S$ which has a neighbor in S and remove u from S.

It is clear that the above algorithm computes an independent set S of G.

(a) Find a (best possible) lower bound on the expected size of S at the end of the algorithm. Your lower bound should be expressed as a function of n and d.

Hint: First compute the expected numbers of nodes in S *and edges in* $G[S]$ *after Step (I) of the algorithm.*

(b) Assume that the above algorithm has running time $T(n)$ and that it computes an independent set of size at least $\frac{n}{5d}$ with probability at least $\frac{1}{2}$.

Show how to compute an independent set of size at least $\frac{n}{5d}$ with probability $1 - \frac{1}{n}$. What is the running time of your algorithm?

Exercise 3: Randomized Partial 3-Coloring *(7 Points)*

The maximum 3-coloring problem asks for assigning one of the colors $\{1, 2, 3\}$ to each node $v \in V$ of a graph $G = (V, E)$ such that the number of edges $\{u, v\} \in E$ for which u and v get different colors is maximized. A simple randomized algorithm for the problem would be to (independently) assign a uniform random color to each node.
What is the expected approximation ratio of this algorithm?

Hint: Consider the approximation ratio to be the minimum ratio of the algorithm solution to the optimal solution over all input instances.

Exercise 4: Max Cut *(1+3+5+3 Points)*

Let $G = (V, E)$ with $n = |V|, m = |E|$ be an undirected, unweighted graph. Consider the following randomized algorithm: Every node $v \in V$ joins the set S with probability $\frac{1}{2}$. The output is $(S, V \setminus S)$.

(a) What is the probability to actually obtain a cut?

(b) For $e \in E$ let random variable $X_e = 1$ if e crosses the cut, and $X_e = 0$, else. Let $X = \sum_{e \in E} X_e$. Compute the expectation $\mathbb{E}[X]$ of X.

(c) Show that with probability at least 1/3 this algorithm outputs a cut which is a $\frac{1}{4}$-approximation to a maximum cut (i.e. a cut of maximum possible size is at most 4 times as large).

Remark: For a non-negative random variable X, the Markov inequality states that for all $t > 0$ we have $\mathbb{P}(X \geq t) \leq \frac{\mathbb{E}[X]}{t}$.

Hint: Apply the Markov inequality to the number of edges **not** *crossing the cut.*

(d) Show how to use the above algorithm to obtain a $\frac{1}{4}$-approximation of a maximum cut with probability at least $1 - \left(\frac{2}{3}\right)^k$ for $k \in \mathbb{N}$.

Remark: If you did not succeed in (c) you can use the result as a black box for (d).