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

Algorithm Theory
Exercise Sheet 12

Due: Friday, 26th of January 2024, 10:00 am

Exercise 1: Modified Contraction *(10 Points)*

(a) Let's modify the contraction algorithm from the lecture in the following way: Instead of contracting a uniform random edge, we choose a uniform random pair of remaining nodes in each step and merge them. That is, as long as there are more than two nodes remaining, we choose two nodes $u \neq v$ uniformly at random and replace them by a new node w. For all edges $\{u, x\}$ and $\{v, x\}$ we add an edge $\{w, x\}$ and remove self-loops created at w.

1. Give an example graph of size at least n where the above algorithm does not work well, that is, where the probability of finding a minimum cut is exponentially small in n (show that in the second part). *(2 Points)*
2. Show that for your example the modified contraction algorithm has probability of finding a minimum cut at most a^n for some constant $a < 1$. *(4 Points)*

(b) The edge contraction algorithm has a success probability $\geq 1/\binom{n}{2}$. We used properties of this algorithm to show that there are at most $\binom{n}{2}$ minimum cuts in any graph. The improved (recursive) min-cut algorithm has a success probability $\geq 1/\log n$. Why can't we use the same argumentation to show that there are at most $\log n$ minimum cuts in any graph (which clearly isn't true as we have seen that cycles have $\binom{n}{2}$ minimum cuts). *(4 Points)*

Exercise 2: Dominating Set in Regular Graphs *(10 Points)*

Let $G = (V, E)$ be an undirected graph. A set $D \subseteq V$ is called a *dominating set* if each node in V is either contained in D or adjacent to a node in D.
We consider the following randomized algorithm for d-regular graphs (i.e., graphs in which each node has exactly d neighbors).

Algorithm 1 `domset`(G)

1: $D = \emptyset$
2: Each node joins D independently with probability $p := \min\{1, \frac{c \cdot \ln n}{d+1}\}$ for some constant $c \geq 1$
3: Each node that is neither in D nor has a neighbor in D joins D
4: **return** D

For simplicity, in all tasks you may assume that $\frac{c \cdot \ln n}{d+1} \leq 1$, i.e., that $p = \frac{c \cdot \ln n}{d+1}$.

(a) Show that the expected size of D (after the execution of `domset`) is at most $\frac{cn \ln n}{d+1} + 1$. *(3 Points)*
Hint: Use the inequality $(1 - x) \leq e^{-x}$.

(b) Show that after line 2 of `domset`, D has size $O\left(\frac{n \ln n}{d+1}\right)$ with probability at least $1 - \frac{1}{n}$. *(2 Points)*

Hint: You might want to use Chernoff's Bound: If $X_1, \ldots, X_n$ is a sequence of independent 0-1 random variables, $X = \sum X_i$ and $\mu = E[X]$, then for any $\delta > 0$ we have

$$\Pr(X \geq (1+\delta)\mu) \leq e^{-\frac{\min\{\delta,\delta^2\}}{3}\mu} .$$

(c) Show that for $c \geq 2$, with probability at least $1 - \frac{1}{n}$, no node joins D in line 3 of `domset`. *(3 Points)*

(d) Conclude that for $c \geq 2$, `domset` returns a dominating set of size $O\left(\frac{n \ln n}{d+1}\right)$ with probability at least $1 - \frac{2}{n}$. *(2 Points)*