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

Algorithm Theory
Sample Solution 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)*

Sample Solution

(a) This algorithm is not efficient. Let (A, B) be a minimum cut. For the edge contraction algorithm we know that it outputs (A, B) if and only if it never contracts an edge crossing (A, B) (chapter 7, part V, slide 8). If the there are k crossing edges, we know that there are $\Omega(k \cdot n)$ edges in the graph and hence the probability to choose a crossing edge is $O(1/n)$ (in the first contraction step). In contrast, for the "node contraction" algorithm, it holds that it outputs (A, B) if it never contracts a "crossing pair", i.e., a pair of nodes $\{a, b\}$ with $a \in A$, $b \in B$, regardless whether there is an edge between a and b. The total number of node pairs is $\binom{n}{2} = \Omega(n^2)$, but the number of crossing pairs can be $\Omega(n^2)$ as well, leading to a constant probability that a crossing pair is chosen.

To formalize this argument, consider the following graph: Let n be even. There are two cliques (= graph with an edge between each pair of nodes) of size $n/2$ and a single edge between these cliques, i.e., an edge $\{u, v\}$ such that u is in the one clique and v in the other, and no more edges between the cliques exist.

So there is a unique minimum cut and we show that the probability that the node contraction algorithm chooses this cut is exponentially small. We only consider the first $n/5$ rounds. In these rounds, there are at least $4n/5$ nodes in the graph, i.e., there are $\binom{4n/5}{2}$ pair of nodes. In order for the minimum cut to survive, the two chosen nodes must be within the same clique. Each clique has size at most $\binom{n/2}{2}$ nodes, i.e., there are at most $2\binom{n/2}{2}$ pairs for which the minimum cut would survive. This yields a probability of at most

$$\frac{2\binom{n/2}{2}}{\binom{4n/5}{2}} = \frac{2\frac{n}{2}\left(\frac{n}{2}-1\right)}{\frac{4n}{5}\left(\frac{4n}{5}-1\right)} = \frac{5}{4}\frac{\left(\frac{n}{2}-1\right)}{\left(\frac{4n}{5}-1\right)} = \frac{\frac{5n}{2}-5}{\frac{16n}{5}-4} \overset{(*)}{<} \frac{\frac{5n}{2}}{\frac{15n}{5}} = \frac{5}{6}\ .$$

$(*)$: For $n > 20$ we have $\frac{n}{5} > 4$ and hence $\frac{16n}{5} - 4 > \frac{15n}{5}$.

It follows that the probability that the minimum cut survives the first $n/5$ rounds is less than $\left(\frac{5}{6}\right)^{n/5} = a^n$ with $a = \left(\frac{5}{6}\right)^{1/5} < 1$, i.e., exponentially small.

(b) In the edge contraction algorithm, we showed that for any minimum cut (A, B), the probability that the algorithm returns (A, B) is $\geq 1/\binom{n}{2}$. As for two minimum cuts $(A, B) \neq (A', B')$, the events "the algorithm returns (A, B)" and "the algorithm returns (A', B')" are disjoint, the probability that the algorithm returns *some* minimum cut is $\geq \frac{\#\text{mincuts}}{\binom{n}{2}}$ and hence #mincuts $\leq \binom{n}{2}$.

In the recursive algorithm, we considered a set S of cuts which are returned by different executions of the edge contraction algorithm and showed that the probability that a specific minimum cut is in S is $\geq 1/\log n$. As for two minimum cuts $(A, B) \neq (A', B')$, the events "(A, B) is in S" and "(A', B') is in S" are not necessarily disjoint, we can not draw any conclusion from the sucess probability to the number of minimum cuts.

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 = ∅
2: Each node joins D independently with probability p := min{1, c·ln n/(d+1)} for some constant c ≥ 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)*

Sample Solution

(a) For every $v \in V$

$$\begin{aligned}\Pr(v \in D) &= \Pr(v \text{ joins } D \text{ in line 2}) + \Pr(v \text{ joins } D \text{ in line 3}) \\ &= \frac{c\ln n}{d+1} + \left(1 - \frac{c\ln n}{d+1}\right)^{d+1} \leq \frac{\ln n}{d+1} + e^{-c\ln n} \\ &= \frac{c\ln n}{d+1} + \frac{1}{n^c}\end{aligned}$$

We obtain

$$E[|D|] \leq n \cdot \left(\frac{c\ln n}{d+1} + \frac{1}{n^c}\right) = \frac{c \cdot n\ln n}{d+1} + \frac{1}{n^{c-1}} \leq \frac{c \cdot n\ln n}{d+1} + 1 \ .$$

(b) For each node v let X_v be the random variable with $X_v = 1$ if v joins D in line 2 and $X_v = 0$ else and let $X = \sum X_v$. We have $\Pr(X_v = 1) = \frac{c \cdot \ln n}{d+1}$ and hence $\mu = E[X] = \frac{cn \cdot \ln n}{d+1}$.

For $\delta = 3$ we obtain

$$\Pr(X \geq (1+3)\mu) \leq e^{-\mu} = e^{-\frac{cn\ln n}{d+1}} \leq e^{-c\ln n} = \frac{1}{n^c} \leq \frac{1}{n}$$

and So with probability at least $1 - \frac{1}{n}$ we have $|D| \leq 4\mu = O\left(\frac{n\ln n}{d}\right)$.

(c) For every $v \in V$, $\Pr(v \in D \text{ in line 3}) = (1-p)(1-p)^d \leq e^{-c\ln n} = \frac{1}{n^c}$.

and $\Pr(\cup_{v\in V} v \in D \text{ in line 3}) \leq \sum_{v\in V} \Pr(v \in D \text{ in line 3}) \leq \frac{n}{n^c} \leq \frac{1}{n}$, for $c \geq 2$, thus with probability at least $1 - \frac{1}{n}$, no node joins D in line 3.

(d) In general, let A, B be two events such that $A \subseteq B$, then $\Pr(A) \leq Pr(B)$.

Hence, $Pr\big($`domset` returns a dominating set of size $O\left(\frac{n\ln n}{d+1}\right)$ at the end of its execution$\big) \geq$ $Pr\big($`domset` returns a dominating set of size $O\left(\frac{n\ln n}{d+1}\right)$ at the end of line 2$\big) \geq (1-\frac{1}{n})(1-\frac{1}{n}) \geq 1-\frac{2}{n}$.