Algorithms Department **Summer 2021**
Department of Computer Science - IBR
TU Braunschweig

Prof. Dr. Sándor P. Fekete
Dr. Phillip Keldenich
Dominik Krupke

Approximation Algorithms
Exercise 3
May 19, 2021

Please hand in your solutions until June 2, 11:30 am by e-mail to `keldenich@ibr.cs.tu-bs.de`.

Exercise 1 (Maximum Cardinality Cut):
Consider the problem Maximum Cardinality Cut, for which we are given an undirected graph $G = (V, E) = (\{v_1, \ldots, v_n\}, E)$. We want to partition V into two disjoint subsets S and T with $S \cup T = V$ such that the number of edges vw between a vertex $v \in S$ and a vertex $w \in T$ is maximized. We call these edges *crossing edges* because they *cross* the cut (S, T).
Consider the following algorithm A. Initially, A sets

$$S = \{v_1, \ldots, v_{\lceil n/2 \rceil}\}, T = \{v_{\lceil n/2 \rceil + 1}, \ldots, v_n\}.$$

In each step, A searches for a vertex v_i that can be moved from S to T or vice versa, increasing the number of crossing edges between S and T by at least 1. If such a vertex v_i can be found, A moves it to the other set and continues with another step. Otherwise, A terminates and outputs S, T.

(a) Argue that A has polynomial running time.

(b) Prove that A is a $\frac{1}{2}$-approximation algorithm.

(c) Give an example that shows that A is not better than a $\frac{1}{2}$-approximation.

(2+9+4 P.)

Exercise 2 (Shortest Substring and Set Cover):
Consider the variant of Shortest Superstring for which we are given a set of strings s_i and want to find the shortest string that contains, for each s_i, either s_i or its reverse s_i^R as a substring. Give an $\mathcal{O}(\log n)$-approximation algorithm for this problem.
Hint: Use greedy set cover. In this variant, a string s_i can be covered either by s_i or s_i^R being present in the output; adapt the sets and the corresponding strings accordingly.
(15 P.)