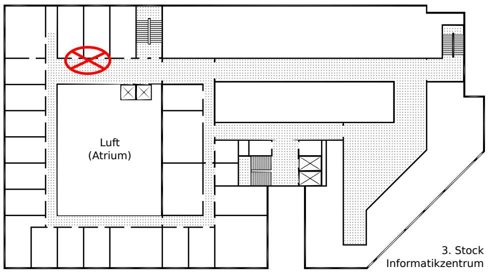
Algorithms Group **Summer '23**
Departement of Computer Science - IBR
TU Braunschweig

Prof. Dr. Sándor P. Fekete
Dr. Ahmad Moradi

Approximation Algorithms: Homework 2
24. May

Solutions are due on June, 7th until 11:00 AM in the cupboard for handing in practice sheets. Please put your name on all pages.



Exercise 1 (Minimum Dominating Set):
A dominating set in a graph $G = (V, E)$ is a set $S \subseteq V$ such that for each $u \in V$, either $u \in S$, or some neighbor v of u is in S. In other words S covers/dominates all the nodes in V. In the Dominating Set problem, the input is a graph G and the goal is to find a smallest sized dominating set in G.

a) Give an example graph for which a dominating set is not a vertex cover and vice versa.

b) Show that Minimum Dominating set problem is a special case of the Minimum Set Cover problem.

c) Adapt the greedy approximation algorithm for set cover we studied in a previous lecture. Is it again an algorithm of logarithmically large approximation factor?

d) Show that the Minimum Set Cover problem can be reduced in an approximation preserving fashion to the Minimum Dominating Set problem.

(5+5+15+10 P.)

Exercise 2 (Maximum Disjoint Paths):
Assume that we are given a directed graph G together with k source-distination pairs of nodes $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$. We think of each pair (s_i, t_i) as a routing request which asks for a path from s_i to t_i. A solution to this problem consists of a subset of requests we will satisfy, $I \subseteq \{1, 2, \ldots, k\}$, together with paths that satisfy them while not overlaping any one edge: a path P_i for $i \in I$ so that P_i goes from s_i to t_i and each edge is used by at most one path. The problem is to satisfy as many requests as possible, i.e. to find a solution with $|I|$ as large as possible. Note that we are requiring the paths to be fully edge-disjoint: $\forall i, j \in I : P_i \cap P_j = \emptyset$.

a) Give an example in which carelessly picking a first path could give a poor solution to the problem. Motivating the following idea.

b) Consider the following algorithmic idea:

Compute shortest path distance between every (s_i, t_i) *pair. Route the one with smallest distance along the corresponding shortest path, remove all the used edges from the graph and repeat.*

Observe that this simple algorithm does not sufffer from what you observed in a).

Show that this algorithm is $O(\sqrt{m})$-approximation algorithm for the problem where m is the number of edges of G. Provide a bad example.

(5+5+10+10 P.)