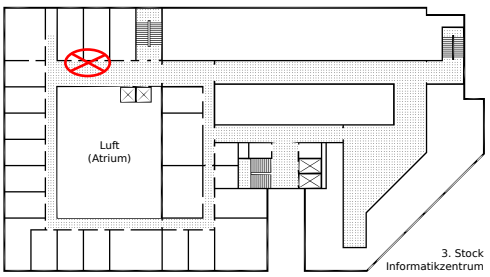
Algorithms Group **Winter '19/20**
Departement of Computer Science - IBR
TU Braunschweig

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

Approximation Algorithms
Exercise 3
27. November 2019

Hand in your solutions until December 11, 11:30 am. You can hand in your solutions at the beginning of the tutorial or via the homework box in front of room IZ337. Please put your name on all pages.



Exercise 1 (Greedy Maximum k-Cut):
In the Maximum k-Cut Problem, we are given an *unweighted* graph $G = (V, E)$. Our goal is to find a partition of V into $k \geq 2$ subsets $V_1, \ldots, V_k$ such that the number of *cut edges* is maximized. An edge $e = v_i v_j$ is a *cut edge* if v_i and v_j are in different components $V_k \neq V_\ell$ of our partition.
Show that there is a simple greedy algorithm achieving an approximation factor of $1 - \frac{1}{k}$. Is this approximation factor tight? **(15 P.)**

Exercise 2 (Maximum Directed Cut):
In the Maximum Directed Cut Problem, we are given a *directed* graph $G = (V, E)$ with non-negative edge weights $w : E \to \mathbb{R}$. The goal is to find a maximum directed cut, i.e., a set $S \subset V$ such that the weight of the outgoing edges of S, i.e.,

$$\sum_{s \in S, t \notin S, st \in E} w(st),$$

is maximized. Give a $\frac{1}{4}$-approximation algorithm for this problem. **(20 P.)**

Exercise 3 (Graph Coloring):
In the Graph Coloring Problem, we are given an undirected graph $G = (V, E)$. Our goal is to find a *k-coloring* $c : V \to \{1, \ldots, k\}$ of G's vertices such that no two adjacent vertices have the same color, i.e., $c(v) \neq c(w)$ if $vw \in E$; we want to minimize the number of colors k.

a) Let Δ be the maximum degree of a vertex in G. Give a greedy algorithm coloring G with $\Delta + 1$ colors.

b) Let G be 3-colorable. Give an algorithm coloring G with $\mathcal{O}(\sqrt{n})$ colors.
Hint: Let $N(v)$ be the (open) neighborhood of a vertex, i.e., all vertices adjacent to v (not including v itself). If G is 3-colorable, the induced subgraph $G[N(v)]$ is bipartite. Use this to remove all vertices of degree $\geq \sqrt{n}$ from G, then continue with the algorithm from a).

(5+10 P.)

Remarks

- These exercises are taken from *Vazirani, Vijay V. Approximation algorithms. Springer Science & Business Media, 2013.*