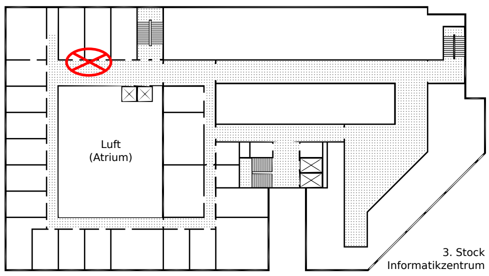
Algorithms Group **Winter '19/20**
Departement of Computer Science - IBR
TU Braunschweig

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

Approximation Algorithms
Exercise 4
10. Dezember 2019

Hand in your solutions until January 8, 2020, 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.
You can reach at most 50 points on this exercise sheet.



Exercise 1 (Approximation Algorithm for Milling):
Given an orthogonal polygon P and a 1×1 square *milling tool* χ, the milling problem asks for a shortest tour T such that moving the center of χ along T covers all points in P without touching any point outside of P.

(a) Show that there is an efficient algorithm deciding whether a given orthogonal polygon can be milled, i.e., whether such a tour T exists.

(b) Assuming that such a tour T exists, provide an efficient $5/2$-approximation algorithm for the milling problem.
Hint: Use a tour covering the boundary of P and a set of horizontal strips that cover the interior of P, combining the two using the ideas from Christofides' algorithm presented in the lecture.

(5+15 P.)

Exercise 2 (Minimum Weight k-Matching):
Let $G = (V, E)$ be a given undirected graph. A k-matching M of G is a subset of the edges of G such that every vertex is incident to exactly k edges in M. Given a non-negative edge cost function $c(e) : E \to \mathbb{R}$, the minimum cost k-matching problem asks for a k-matching M of G with minimum possible cost $\sum_{e \in M} c(e)$.
The minimum weight 1-matching problem can be solved in polynomial time by some algorithm $\mathcal{A}$. Show that $\mathcal{A}$ can be used to solve the minimum weight k-matching problem in polynomial time for any k.
Hint: Replace every vertex v in G by $2\deg(v) - k$ vertices. **(20 P.)**

Exercise 3 (1-2-Graph TSP):
Let $G = (V, E)$ be an undirected *complete* graph with non-negative edge costs $c(e)$ with $c(e) \in \{1, 2\}$ for all edges e. Provide an efficient $^4/_3$-approximation algorithm for the TSP on G.
Hint: Begin with a minimum cost 2-matching, which covers G by a minimum weight set of cycles. **(20 P.)**