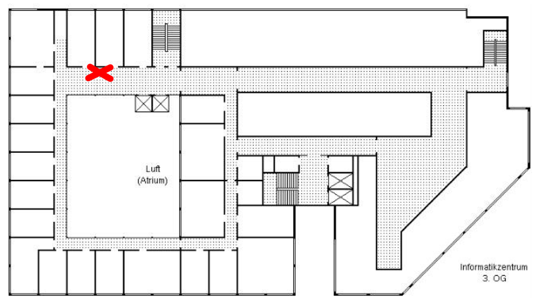
Abteilung Algorithmik
Institut für Betriebssysteme
und Rechnerverbund
TU Braunschweig

SoSe 15

Prof. Dr. Sándor Fekete
Dr. Victor Alvarez
Melanie Papenberg

Approximation Algorithms
Homework Set 3, 01. 06. 2015

Solutions are due on Monday, June 15th, 2015, until 13:00 in the cupboard for handing in practice sheets. **Please put your name on all pages!**



Exercise 1 (Approximating the minimum-degree spanning tree of a graph in polynomial time): In the big tutorial of the 21st of May, 2015, *one* approximation algorithm for the minimum-degree spanning tree of a Graph $G = (V, E)$ was presented. In this exercise you will prove in a series of steps that the algorithm runs in polynomial time. To ease computations, assume that in each round the algorithm tries to reduce the degree of a vertex v such that $d_T(v) \geq \Delta(T) - \log_2 n$, where $\Delta(T)$ denotes the maximum degree of the current solution T, and $d_T(v)$ denotes the degree of v in T.

For a spanning tree T consider the potential function $\Phi(T) = \sum_{v \in V} 3^{d_T(v)}$.

1 What is an upper bound for the highest (initial) potential? What is the lowest potential that can be achieved?
2 Show that after *each* round (phase) of the algorithm, the potential function decreases by a factor of at least $\frac{2}{27n^3}$
3 Consider the situation after $\frac{27}{2}n^4 \ln 3$ rounds and conclude that at that point the resulting tree must be locally optimal. That is, after that many rounds no more valid moves can be performed any longer.

Along the way you may use the inequality $1 - x \leq e^{-x}$. **(5 + 10 + 5 = 20 pts.)**

Exercise 2 (Greedy packing of $k \times k$ squares): Show that if we greedily pack $k \times k$ squares into a polyomino, we obtain a 4-approximation (for the problem of packing the largest number of squares into a polyomino). **(5 pts.)**.

Exercise 3 (Running time of the shifting technique): The shifting technique of Hochbaum and Maass was presented in class, without full details of the running time. Provide a formal analysis of its running time. Assume you are packing $k \times k$ squares, with $k \in \mathbb{N}$.
Hint: For this problem you first have to prove that there is an optimal solution (packing) in which *each packed* square has its vertices at grid points. **(20 pts.)**

Exercise 4 (Shifting Technique in d dimensions): Generalize the algorithm presented in class for two dimensions to general $d \geq 2$ dimensions. Provide the corresponding analysis of the approximation factor. You may assume that you are packing 2×2 squares. **(5 + 10 = 15 pts.)**