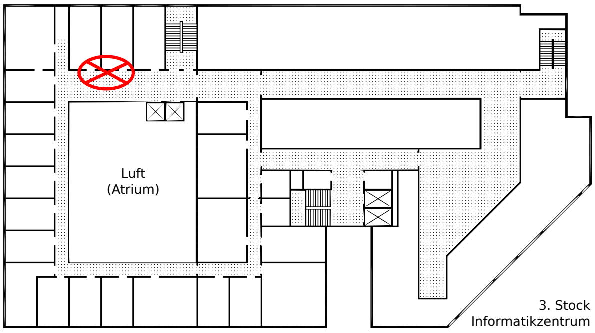
Abteilung Algorithmik **Summer term 2016**
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig

Prof. Dr. Sándor P. Fekete
Phillip Keldenich

Online Algorithms
3rd Homework Assignment, 25th of May, 2016



Solutions are due on Wednesday, the 8th of June, 2016, until 11:30 in the homework cupboard. You can also hand in your solution in person before the small tutorial begins. External students can hand in the homework via e-mail to `c.rieck@tu-bs.de`.

Please make sure your homework submissions are clearly labeled using your name and/or matriculation number.

Exercise 1 (Line search): Suppose that you are a blind hungry cow at some position on a barren path. You know that, by following the path in one of the two possible directions, you will eventually come across a pasture with tasty grass. Once you step on the pasture, you will immediately recognize it, but no earlier. Assume that you can accurately follow the path, change direction and measure the distance travelled.

Let $D \geq 1$ be the distance from the point of origin to the pasture. In the lecture, we outlined a strategy to find the pasture and stated that the distance travelled using this strategy is no more than 9 times the distance D. Prove that the strategy outlined in the lecture achieves a competitive factor of 9. Also prove that this bound is tight, i.e. that the strategy is not c-competitive for any constant $c < 9$. **(15 points)**

Exercise 2 (Dynamic monotone trees): In the tutorial, we presented the dynamic binary search tree algorithms Rotate-To-Root, Single-Rotation and Splay. All these algorithms have in common that they do not use additional memory compared to a regular binary search tree. Such strategies are also called *stateless*.

In this exercise, we consider a generalization of the Frequency-Count list update algorithm to trees. In a so-called *dynamic monotone tree*, each element stores a counter that maintains how often the element has been searched for. The counters are all initialized to 0. Whenever we search for an element, we increase its counter and rotate it up until its counter is not bigger than the counter of its parent. In other words, we maintain a search tree which is also in max-heap order with respect to the access counters.

Prove that dynamic monotone trees are not better than $\Omega(n)$-competitive. **(20 points)**

Exercise 3 (Graph exploration): In this exercise, we consider the problem of exploring a strongly-connected directed graph. Each vertex is labeled by a natural number $i \in \{1, \ldots, n\}$. We do not know the incoming and outgoing edges of a vertex before visiting it. We start at vertex 1 and initially know the outgoing edges of 1. The goal is to visit each vertex at least once and return to the start by following the directed edges according to their direction. Traversing each edge incurs a cost of 1.

a) Prove that no deterministic online algorithm has a competitive ratio better than $\frac{n+1}{2} - \frac{1}{n}$.

b) Devise a strategy that has competitive ratio $\frac{n+1}{2} - \frac{1}{n}$ and prove this upper bound.

(10+15 points)