Online Motion Planning, SS 17
Exercise sheet 1
University of Bonn, Inst. for Computer Science, Dpt. I

- *You can hand in your written solutions until Wednesday,* 25.04., 14:15*, postbox in front of room E.01 LBH.*
- *We allow (and recommend) fixed groups of 2 students.*
- *Please subscribe to our mailing list:* *https://lists.iai.uni-bonn.de/mailman/listinfo.cgi/vl-online*

Exercise 1: Basics of Competitive Analysis (4 points)

In this exercise a robot is placed in some grid cell inside an unknown grid graph G. The robot performs an exploration strategy ALG until it has found some special item, which is located at a target destination (which is also unknown to the robot). Prove the following statements assuming that $k \in \mathbb{N}$ is a constant.

a) For *any* (correct) exploration algorithm ALG, if G is (unknown to the algorithm) a grid graph of at most k cells, then for *any* constant $c > 0$, it holds that there exists a constant α where

$$\mathrm{ALG} \leq c \cdot \mathrm{OPT} + \alpha$$

b) Prove that if the size of G is not necessarily bounded, then for any constant $1 > c > 0$ there is no algorithm ALG for which exists a constant α such that

$$\mathrm{ALG} \leq c \cdot \mathrm{OPT} + \alpha$$

holds.

Please turn the page!

Exercise 2: The MOUSE Algorithm (4 points)

We consider the *MOUSE* Algorithm for finding a destination grid cell inside an unknown grid graph G. Suppose graph G is a horizontal corridor consisting of n cells, as shown in Figure 1. Initially, in each cell, the mark used by

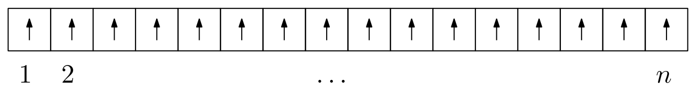

Figure 1: A vertical corridor of length n.

the algorithm points in direction *NORTH*.

Assuming that the destination grid cell is cell 1, answer the following questions.

a) How many moves are made by the *MOUSE* Algorithm before it finds the destination cell, depending on the starting cell i, $1 < i \leq n$? Starting from which cell maximizes this value?

b) Let *MOUSE*(i) denote the number of steps required to find the target, starting from cell i, as determined in part *a*). Which starting position maximizes the value

$$\frac{\text{MOUSE}(i)}{\text{OPT}(i)}$$

where $\text{OPT}(i)$ denotes the number of steps required by an optimal offline strategy *OPT*, starting in cell i.

Exercise 3: ***MOUSE* using *DFS*** **(4 points)**

In order to improve the performance of the *MOUSE* Algorithm we want our mouse to explore the given grid graph using the *DFS* Algorithm. To this end, we enhance the capabilities of the mouse as follows.

First, instead of one mark each cell now has four different marks, each of which the mouse can distinguish (by its smell). Second, the mouse can now read the marks of adjacent cells, that is, of the adjacent cells in *NORTH, EAST, SOUTH* and *WEST* direction of its current cell.

Describe how the mouse can find an arbitrary destination cell by performing the *DFS* Algrorithm. It is not required to provide *(PSEUDO-)Code.* Instead, focus on how the different marks can be used to represent the information used by the *DFS* Algorithm – and how they are updated during the execution of the algorithm.

Hint: Be careful not to use extra memory the mouse does not possess – e.g. for a recursion stack. The mouse knows its algorithm and makes decisions based on (the algorithm and on) the marks on the floor only.

Hint: Recall that – although the mouse does not possess a compass – after moving to a cell it knows that it has just moved and that it faces in direction of its last movement.