University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
S. Faour, M. Fuchs, A. Malyusz

Algorithm Theory
Sample Solution Exercise Sheet 11

Due: Friday, 19th of January 2024, 10:00 am

Exercise 1: Robot in 1D *(3+7 Points)*

We have a robot on the 1D line at the 0th position additionally you know that at some distance integer $D \geq 1$ along the path there is your car; however, you do not know the distance D or the direction in which you have to go. The robot can move 1 integer step in any direction. If the robot is at a point on the line you automatically check if the given integer point has the car or not. Your objective is to minimize the number of steps the robot takes to reach the car. Create an algorithm that has

(a) 16 competitive ratio. (You will move at most $16D$.)

(b) 9 competitive ratio.

(If you give an algorithm with 9 competitive ratio then you automatically solved problem a too.)

Sample Solution

Let our algorithm be the following. In order to reach the car, you start by traveling into one of the two directions for one unit of distance. If you do not find your food, you return to the start and then travel two units of distance into the other direction. If you still have not found your food, you return to the start and then travel four units of distance into the first direction, and so on. You repeat this procedure, doubling the distance each time you return back to the start, until you finally reach your goal. We now show that this algorithm has competitive ratio of 9. In phase i robot visits location $(-2)^i$ and travels the distance $2 \cdot 2^i$. Worst case is when an object is located just outside of the radius covered in some phase. Then the robot returns to the origin, doubles the distance and travels in the "wrong direction", returns to the origin, and discovers the object by travelling in the "right direction." In other words when an object is at distance $d = 2^i + \epsilon > 2^i$ in direction $(-1)^i$, the total distance travelled is $2(1 + 2 + \cdots + 2^i + 2^{i+1}) + d \leq 2 \cdot 2^{i+2} + d < 8d + d = 9d$. Thus, this doubling strategy gives a 9-competitive algorithm for the line search problem.

Exercise 2: Online MST *(10 Points)*

Let G be a complete graph on n nodes. Every edge has a weight in the interval of $[1, 2]$. Our aim is to get an MST. The edges with the weights arrive one by one, and we need to decide to include the current edge or not to in the spanning tree. Create an online-algorithm with competitive ratio of $\sqrt{2}$. *Hint: Use a carefully chosen global constant c which you use to decide what to do with a given edge. Is the weight of the edge smaller or larger than c.*

Sample Solution

Let us take the following general algorithm. For the algorithm we will use a properly chosen constant c. If the current edge can not be added to the current spanning forest we skip it. If we remove the

current edge from the graph and the component number increases then we include this edge in the spanning tree. Otherwise, if the weight of the edge is $\leq c$ we include it in the spanning tree else we skip it. To prove that we have competitive ratio of $\sqrt{2}$ we create a bijection between the edges of the output of the algorithm and an optimal solution. Let us go through every cut the spanning tree edges span, and we can see the following:
If the output edge has weight $\geq c$ that means we included it because there the component number would have increased means every edge in the cut had at least c weight. Otherwise, the weight of the edge is $\leq c$. This means that for every edge in the output and its image in the optimal solution one of the following statements is true either both of them has $\leq c$ weight or $\geq c$.
If we choose $c = \sqrt{2}$ we get that we have $\sqrt{2}$ competitive ratio. The competitive ratio for a given edge can be bounded if we take the max weight for the output edge and the least for the optimal edge. This means for example if both of them has weight $\leq c$ that $w_{outputedge} = c, w_{optimaledge} = 1$ and thus the ratio is $\sqrt{2}$.