Prof. Dr. Sándor P. Fekete
Peter Kramer

Due: 24.06.2024
Discussion: 01.07.2024

Sheet 4

Please submit your individual solutions using the boxes in front of IZ338, before the exercise timeslot on the due date above. Your homework submission may be handwritten using proper ink (no pencil, no red ink) or printed.

Exercise 1 (Online Exploration: Cow Path Problem): **(15+5 points)**
Suppose that you are a hungry cow on a path. You know that at some distance $D \geq 1$ along the path there is a pasture with tasty grass; however, you do not know the distance D or the direction in which you have to go.

Consider the following strategy: Start by traveling into one of the two directions for *one* unit of measurement. If you do not find your food, you return where you started and then travel *two* units into the other direction. If you still have not found the pasture, you again return to the start and travel *four* units of distance into the first direction, and so on. Repeat this procedure, doubling the distance each time you return back to the start, until you finally reach your goal.

a) Prove that this strategy is 9-competitive, i.e., that you never have to travel more than $9D$ units of distance until you reach the goal. Moreover, prove that this is tight, i.e., that there is no constant $c < 9$ such that this strategy is c-competitive.

b) Prove that the restriction $D \geq 1$ is vital, i.e., that without it, there is no c-competitive strategy for any constant c.

Exercise 2 (List Update Problem: MoveToFront): **(Bonus: 10+15 points)**
In this exercise, we consider the List Update Problem:

We are tasked with maintaining a set of keys in a linked list and are given a sequence of queries σ for keys. For each query, we iterate through the list, starting from the first element. Each element we access in our search costs us 1 unit. Searching for 2 costs 1 in $[2, 1, 4, 3]$, costs 2 in $[1, 2, 3, 4]$ and costs 4 in $[1, 3, 4, 2]$ units. After finding the queried element, we can move it to any point closer to the front *without* extra cost. After finding 2 in $[1, 3, 2, 4]$, which costs 3 units, we may thus change the list to $[2, 1, 3, 4]$ or $[1, 2, 3, 4]$ or keep $[1, 3, 2, 4]$.

The MoveToFront algorithm acts as follows: After each request s_i, move the requested item s_i to the front of the list. For example, after searching for 2 in $[1, 4, 2, 3]$, the list becomes $[2, 1, 4, 3]$.

a) Prove that there is no constant $c < 2$ such that MoveToFront is c-competitive.

b) Prove that MoveToFront is 2-competitive.

(Hint: Use the number of inversions in MoveToFront's list w.r.t. OPT's list after request i as a potential function $\phi(i)$. An inversion is a pair x, y of elements such that x comes before y in MoveToFront's list, but after y in OPT's list.)

In order to prove $c_{\mathrm{MTF}}(i) + \phi(i) - \phi(i-1) \leq 2c_{\mathrm{OPT}}(i)$ for a request s_i, consider the number of items k that come before s_i in both OPT's and MoveToFront's list, the number of items m that come before s_i in OPT's list but after s_i in MoveToFront's list, and the number of items ℓ that come before s_i in MoveToFront's list but after s_i in OPT's list.

Exercise 3 (List Update Problem): **(Bonus: 20 points)**
In the previous exercise, we considered the list update problem and showed that the algorithm MoveToFront is 2-competitive. Prove that no deterministic list update algorithm can be c-competitive for any $c < 2$.