University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
P. Bamberger, P. Schneider

Algorithms Theory
Sample Solution Exercise Sheet 3

Due: Tuesday, 24th of November 2020, 4 pm

Exercise 1: Interval Scheduling *(10 Points)*

Consider the interval scheduling problem from the lecture. We have seen that successively choosing the shortest available interval does not yield an optimal solution. In this task we want to prove that this approach at least gives a *2-approximation* of an optimal solution:

If S is the set of intervals given by the "shortest available interval"-algorithm and O an optimal solution, show that $|S| \geq \frac{|O|}{2}$.

Sample Solution

Each interval in O intersects with at least one interval in S, because if some $I \in O$ had no intersection with an interval in S, then I would still be available after the execution of the "shortest available interval"-algorithm, a contradiction. Hence, if we take for each interval in S the intervals from O it intersects with, we have listed all intervals in O. So it remains to show that an interval I in S intersects with at most two intervals in O. For a contradiction, assume it intersects with three intervals $I_1, I_2, I_3 \in O$ with $a_1 < a_2 < a_3$ (the starting points of the intervals). It follows that I_2 is shorter than I and was available when the algorithm picked I instead, a contradiction.

Exercise 2: Storing Files *(10 Points)*

Assume you have n files with sizes $s_1, \ldots, s_n$ that you want to store on a magnetic tape. You can not directly access a file. Instead, for reading file k, you must go from the beginning of the tape to the place where file k is stored. Hence, accessing file k costs

$$cost(k) = \sum_{i=1}^{k} s_i \ .$$

Assume that all files are accessed equally often. Give an efficient algorithm that stores the files on the tape in an order that minimizes the *average accessing time*. Prove that your algorithm is correct.

Sample Solution

We store the files by size in increasing order. To see that this is optimal, consider an ordering where for some files i and j we have $s_i > s_j$ and file i is directly followed by file j. We show that this ordering is not optimal. When we exchange the files i and j, we obtain new costs for all files. For file i we have

$$cost_{new}(i) = cost_{old}(i) + s_j$$

and for file j

$$cost_{new}(j) = cost_{old}(j) - s_i \ .$$

For $k \notin \{i, j\}$ we have

$$cost_{new}(k) = cost_{old}(k)$$

Hence we obtain

$$\begin{aligned}\sum_{k=1}^{n} cost_{new}(k) &= cost_{new}(i) + cost_{new}(j) + \sum_{k \in \{1,\ldots,n\} \setminus \{i,j\}} cost_{new}(k) \\ &= cost_{old}(i) + s_j + cost_{old}(j) - s_i + \sum_{k \in \{1,\ldots,n\} \setminus \{i,j\}} cost_{old}(k) \\ &= \sum_{k=1}^{n} cost_{old}(k) + s_j - s_i < \sum_{k=1}^{n} cost_{old}(k)\end{aligned}$$

We see that exchanging files i and j reduces the sum of all costs and hence the average cost. It follows that the chosen ordering (with file i before j) was not optimal.