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

Algorithms Theory
Exercise Sheet 2

Due: Monday, 19th of November, 2018, 14:15 pm

Exercise 1: Majority and Frequent Elements *(4+6 Points)*

Let us assume that A is an arbitrary array of size n that contains n integers. An element in A is called a *majority element* if the number of occurrences of the element in A is strictly greater than $n/2$.

(a) Provide an algorithm that returns the majority element of A if there is one. The algorithm must have a time complexity of $O(n)$ and constant auxiliary space.

(b) Argue the correctness of your answer.

Note: Auxiliary space is the space the algorithm needs in addition to the input (i.e., array A) throughout the algorithm execution.

Exercise 2: Covering Unit Intervals *(3+6 Points)*

We are given a set X of real numbers. The problem is to find the minimum cardinality set of unit intervals $S = \{[s_1, s_1 + 1], [s_2, s_2 + 1], \ldots, [s_k, s_k + 1]\}$, where $s_1, s_2, \ldots, s_k \in \mathbb{R}$, such that every real number in X belongs to at least one interval in S.

(a) Consider the following greedy algorithm $\mathcal{A}$, determine whether it solves the problem or not and explain why.

$\mathcal{A}$ proceeds in steps until X becomes empty. In the j^{th} step, $\mathcal{A}$ determines $s_j \in \mathbb{R}$ such that the unit interval $[s_j, s_j + 1]$ contains the maximum number of elements left in X. Then, to conclude the j^{th} step, $\mathcal{A}$ removes all the real numbers from X that are contained in $[s_j, s_j + 1]$.

(b) Provide an *efficient* greedy algorithm to solve the problem. Argue the correctness of your answer.

Exercise 3: Scheduling *(4+7 Points)*

We are given n jobs $J_1 = (s_1, p_1), \ldots, J_n = (s_n, p_n)$, where s_i is the earliest start time and p_i is the processing time of job J_i. The goal is to schedule these jobs on a single processor, whereas each job can be suspended and resumed again as many times as needed. For example a job $J_k = (4, 5)$ could be scheduled to be processed in intervals $[4, 6], [10, 12]$, and $[15, 16]$.

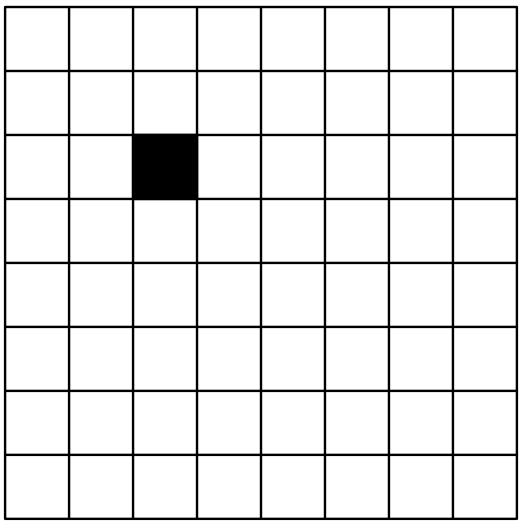
(a) Provide an efficient greedy algorithm to compute a scheduling of the n jobs on a single processor while minimizing parameter $C = \sum_{i=1}^{n} c_i$, where c_i is the time when job J_i is completed.

(b) Argue the correctness of your answer.

Exercise 4: Divide and Conquer *(2+6+2 Points)*

Consider a $n \times n$ *square* grid with $n = 2^k$ for a $k \in \mathbb{N}_{\geq 1}$. We have an unlimited supply of a specifically shaped tile, which covers exactly 3 cells of the grid as follows:

The goal is to cover the whole grid with these tiles (which can also be turned by 90, 180 and 270 degrees). We call an arrangement of tiles on grid cells a *valid tiling*, if all cells of the $n \times n$ grid can be covered with the tile above *without any overlaps* of tiles and *without going over the edges* of the grid. Assume that the input grid has an arbitrary *single* cell that is initially tiled (before the start of the algorithm). E.g. for $n = 8$ the input grid may look like this:



(a) Is there a *valid tiling* for every $2^k \times 2^k$ grid ($k \in \mathbb{N}_{\geq 1}$) that is initially completely empty? Prove or disprove.

(b) Describe a *divide and conquer* algorithm that computes a *valid tiling* on a $n \times n$ grid in $O(n^2)$ (with $n = 2^k, k \in \mathbb{N}_{\geq 1}$) that has one cell that is initially tiled. Assume that placing a tile is in $O(1)$.

(c) Show the running time of $O(n^2)$.