Homework Assignment 3

Due Date: March 7, 2021, 23:59

Note. Please note that this semester all assignments are group assignments. Further note that for the grading we will apply a "10%" rule, i.e. the maximum number of points for this assignments is 55, but 50 will be counted as 100%. Points that exceed 50 will be stored in a separate counter and used later for compensation of lost points in other assignments or (if not used up this way) the final exam.

Exercise 1.

(i) Show that any comparison-based algorithm for determining the smallest of n elements requires $n-1$ comparisons.

(ii) Show also that any comparison-based algorithm for determining the smallest and second smallest elements of n elements requires at least $n-1+\log n$ comparisons.

Note. You must consider that an arbitrary *algorithm* contains these number of comparisons, whereas on a specific input the number may still be lower (see the proof on the number of comparisons in a sorting algorithm).

(iii) Give an algorithm with this performance.

total points: 15

Exercise 2.

(i) Implement max-heaps using arrays. In particular, implement *build_heap* and *sift-down*.

(ii) Implement heapsort using max-heaps.

total points: 14

Exercise 3.

(i) Show how addressable priority queues using doubly linked lists can be realised, where each list item represents an element in the queue, and a handle is a handle of a list item.

(ii) Determine and the complexity of queue operations for two different options using sorted lists or unsorted lists.

total points: 12

Exercise 4.

(i) Design an algorithm for inserting k new elements into a max-heap with n elements.

(ii) Give an algorithm with time complexity in $O(k + \log n)$.

Hint. Use an approach similar to the building of a heap.

total points: 14