Homework Assignment 3

Due Date: March 11, 2022, 23:59

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: 12

Exercise 2. Design an algorithm to find both the largest and the smallest elements in a list with n elements such that at most $2n-3$ comparisons are needed. You may assume that n is a power of 2.

(i) Determine the exact number of comparisons of your algorithm.

(ii) How does the number of comparisons change, if n is not a power of 2?

total points: 10

Exercise 3. Consider a list L with n elements from a totally ordered set T. A *majority element* is an element $x \in T$ such that there are more than $n/2$ list elements x.

Design an algorithm to decide in linear time, whether L contains a majority element. If such an element exists, the algorithm shall return it as the result.

total points: 8

Exercise 4.

(i) Implement a rotation operation *rotate*(m) by m positions (with $m \in \mathbb{N}$) on lists. Use a divide & conquer algorithm that works in-place.

(ii) Implement a selection operation *select*(k) (with $k \in \mathbb{N}$) on lists to find the k'th smallest element of the list. Use a divide & conquer algorithm.

Programming instructions. Use the class template ALIST and add member functions *rotate* and *select*.

Testing instructions. Test your programs on a few lists containing at least 500 random values. Explore different values for m and k including the case of finding the median. Use `append` to create the list.

total points: 20