DUA:

Konstantin Krasser

October 23, 2024

Aufgabe 1 (9 points). In the lecture, you learned about the partition function. This function can not only be used to sort an array A (as in quicksort), but also to find the k-th smallest value in A, i.e., the k-th element in the ascending sorted order of the elements in A. Example: In the array $A = [6, 7, 2, 9, 3, 1, 0]$, the 4 th smallest element is the number 3 (only 0,1 , and 2 are smaller). We assume that every element in A is unique.

1. Answer to question 1:

   What would a naive approach, using comparison-based search algorithms, look like to find the k-th smallest element in A ? What is the lower asymptotic bound on the runtime of this approach?

   The lower bound for any comparison-based algorithm is: The Naive approach would be to first sort the array and then return the $k - th$ element. Thus making the lower bound for:

   $$(O)n \log n \tag{1}$$

2. Answert to question 2:

   (2 points) Provide a modified partition function in pseudocode that rearranges the array A such that the pivot element is at position i_p, and all elements to the left of i_p are smaller than the pivot element, and all elements to the right of i_p are greater than the pivot element. The manipulation of A is done in-place, so all changes are made directly in A, and partition does not need to explicitly return the array A, only i_p.

3. Answer to question 3:

   (4 points) Describe in words and in pseudocode how the modified partition function can be used to efficiently find the k-th smallest value in A.

4. Answer to question 4:

   (2 points) What is the runtime of your algorithm in the best case and in the worst case? Provide an example call for both cases. The best/worst case should apply to general k, not a specific k.