

# **Data Structures**

PSS 3

February 10, 2024

**Prologue meme of the day:**



# Sorting Problems

## Theoretical questions:

0) Suppose we have an array of  $n$  elements:

- a) all equal to each other;
- b) in increasing order
- c) in decreasing order

Specify the time complexity of all the sorting algorithms on these specific arrays.

|                     | Worst Case | Best Case |
|---------------------|------------|-----------|
| Quicksort           |            |           |
| In-Place Quick-Sort |            |           |
| Insertion sort      |            |           |
| Merge sort          |            |           |
| Selection sort      |            |           |
| Bubble sort         |            |           |

1) We have a sequence  $a, b, c, a, b, c, \dots$  where  $a < b < c$  (in total  $3n$  elements). Specify the time complexity of QuickSort algorithm on this sequence.

2) Let  $a, b$  and  $c$  be integers such that  $a < b < c$  and  $n \geq 1$ .

Denote by  $S(n, n, 2n)$  the set of all the sequences composed of keys  $a, b$  and  $c$  in which  $a, b$  occur  $n$  times and  $c$  occurs  $2n$  times.

Choose a sequence for which QuickSort will run with its worst case and best case if the pivot is chosen as the first element

3) Suppose we modify the deterministic version of the quick-sort algorithm so that, instead of selecting the last element in an  $n$ -element sequence as the pivot, we choose the element at index  $\lfloor n/2 \rfloor$ . What is the running time of this version of quick-sort on a sequence that is already sorted?

# **When someone uses bubble/insertion/selection sort instead of QuickSort!!!**



4) Suppose we have an  $O(n)$  time algorithm that finds the median of an unsorted array. Now consider a QuickSort implementation where we first find the median using the above algorithm, then use the median as a pivot. What will be the worst-case time complexity of this modified QuickSort?

5) Consider a situation where changing the position of any element is a very costly operation. Which of the following sorting algorithms should be preferred so that the number of these operations are minimized in general?

insertionSort      selectionSort      bubbleSort

6) Suppose we are sorting an array of integers using in-place-quickSort, and we have just finished the first partitioning with the array looking like this:

a) 9 11 8 12 13 16 18 21 22 21 25 40 33 56 123

Which numbers can be pivot?

b) 11 7 3 5 13 17 14 29 23 31 35 42 78 112 111

Which numbers can be pivot?

7) Which of the sorting algorithms is efficient to sort array consisting of ASCII characters?

## Coding:

0) Suppose we are given two  $n$ -element sorted sequences  $A$  and  $B$  each with distinct elements, but potentially some elements that are in both sequences. Implement an  $O(n)$  time method for computing a sequence representing the union  $A \cup B$  (with no duplicates) as a sorted sequence.

- 1) Implement Bucket Sort
- 2) Implement Radix Sort based on the Bucket Sort

