

# PS Algorithms and Data Structures 2026

## Task sheet 4

### Task 10

For each of the following functions, determine whether the master theorem from the lecture can be applied or not and justify your decision. If possible, also provide an estimate using the master theorem.

1.  $T_1(n) = 81T_1\left(\frac{n}{3}\right) + 3n^4 + n^3$
2.  $T_2(n) = 20T_2\left(\frac{n}{5}\right) + 22T_2\left(\frac{n}{5}\right) + 5n^2 + 8$
3.  $T_3(n) = T_3\left(\frac{n}{2}\right) + \log n$

### Task 11

What (tight) runtime complexity does Quick-Sort have for arrays that are constructed as follows? Justify your answer.

1. Let  $n$  be an odd number.
2. The first  $\frac{n-1}{2}$  numbers of the array are the numbers  $\frac{n-1}{2}, \dots, 1$  in this order.
3. The remaining numbers of the array are the numbers  $n, \dots, \frac{n+1}{2}$  in this order.

Consider the variant of Quick-Sort presented in the lecture, in which the last entry of the current subarray is always used as the pivot element.

### Task 12

Let  $A[1, \dots, n]$  be an array with  $n$  different numbers. If  $i < j$  and  $A[i] > A[j]$  holds for indices  $i$  and  $j$ , then this is called a *Inversion*.

Formulate an algorithm in pseudocode to calculate the number of inversions in an array. The worst-case runtime should be  $O(n \log n)$ .

Tip: Modify one of the sorting algorithms from the lecture in a suitable way.