Alessandro Artale, Werner Nutt,
Ognjen Savković, Francesco Sportelli

6. Algorithms on Arrays

In this lab, you are going to develop efficient algorithms for several elementary problems involving arrays of integers. For each algorithm, you should proceed by the following steps:

1. Write down in words your idea for the algorithm.
2. Write up your algorithm in pseudocode.
3. Determine the asymptotic complexity of your algorithm.

Note that you are not asked to write any Java code. Imagine you want to build a house. Then you do not start digging a hole and hope that the the right ideas about what to do next will come in time. Therefore, the goal of the lab is to let you think about algorithms, i.e., on what one should do before starting to code.

1. Minimum in a Rotated Sorted Array

Consider an array A consisting of distinct values that has first been sorted and then been rotated by some unknown distance $d \geq 0$. For instance, the array $A = [11, 14, 17, 2, 4, 5, 7, 9]$ has been obtained from $[2, 4, 5, 7, 9, 11, 14, 17]$ by a rotation by $d = 3$. You can also think of a rotation as a right shift by d, after which all the elements pushed out of the array are again fed into the left hand side.
The algorithm to be developed in this exercise receives as input a rotated sorted array, where the rotation distance is unknown. The algorithm returns the index of the minimum.
Develop first a recursive algorithm for this task. Then translate the recursive algorithm into an interative one.

2. Intersection of Arrays

Suppose A and B are integer arrays without duplicates. We want to compute a third array C that contains those numbers appearing both in A and in B.
First, write up a straightforward solution. Then try to find a more clever one that uses an algorithm you have seen in the lecture. You need not write up that algorithm again.

3. K-th Smallest Element in an Array

We want to develop an algorithm that receives as input an arbitrary array A and a number K with $1 \leq K \leq A.\mathrm{length}$ and returns the K-th smallest element in the array. We also count multiple occurrences of the same element to arrive at K.
For instance, for the input $A = [9, 5, 2, 7, 7, 4, 5]$ and $K = 4$ the output should be 5, since in the sorted version $[2, 4, 5, 5, 7, 7, 9]$ the number at position 4 is 5.
First, write up a straightforward solution. Then try to find a more clever one that uses Lomuto's partitioning algorithm.