Series No. 01 – Complexity Analysis

Exercise 01: Search in a Circular Array

A circular sorted array is a sorted array in which elements are circularly shifted. For example, the array [12, 14, 18, 21, 3, 6, 8, 9] is a circularly sorted array (in ascending order). The element "3" (at index 4) is called a rotation point (or the pivot) of the array. The goal of this exercise is to write a function that takes a circularly sorted array and an element as parameters and returns the index of the element in the array. If the element is not in the array, the function returns -1.

1. Start with the naive solution which consists of linearly traversing the array until the element is found, or reaching the end of the array. The complexity of this solution is linear ***(O(n))***.
2. It is also possible to use binary search, since the elements of the array are sorted. However, it needs to be adapted to work with a circularly sorted array. Therefore, you should:
    a. Find the rotation point.
    b. Apply binary search on an array that starts from the rotation point to the element just before it.

Note that it is also necessary to use binary search to find the rotation point for this solution to be useful.

Exercise 02: Subarray with Maximum Sum (“Solve it” Competition - 2023)

Write a function that takes an array of integers and returns the maximum sum of a subarray from this array. For example, if the input array is: ***A = [1, 2, -4, 3, 2, -1, 3, -1]***, the function should return ***7***, which is the maximum sum of the subarray ***[3, 2, -1, 3]***. There is no subarray of consecutive elements in A that has a larger sum than ***7***.

1. The naive solution consists of calculating the sum of all possible subarrays, comparing them, and finding their maximum. Provide this solution. What is its complexity?
2. Provide a more efficient solution with linear complexity ***(O(n))***.

Exercise 03: Search in a Sorted Matrix

Given a matrix of integers with n rows and m columns sorted such that:

- The rows are sorted in ascending order.
- The first element of each row is greater than the last element of the previous row.

Write a function ***"search2D"*** that takes a sorted matrix and an integer as parameters and returns true if the integer is present in the matrix and false otherwise.

1. Initially, implement the naive version.
2. It is possible to optimize your solution by using binary search first to find the (possible) row of the searched value. Then use binary search a second time to determine if the searched value exists or not.

Exercise 04: Median of Two Arrays

The median of an array is the element in the middle when it is sorted. If the array contains an odd number of elements, the median will be the central element. If the array contains an even number of elements, the median is defined as the average of the two central elements.

1. Write a function "median" that takes a sorted array A with its size as a parameter and returns the median of this array. The function assumes that the array is already sorted.
2. Write a function "median2" that takes two sorted arrays A and B of sizes n and m respectively and returns the median of these two arrays.
    a. The naive solution would be to merge the two arrays, then find the median of the result.
    b. It is also possible to find the median without merging the arrays, simply by iterating through their elements in ascending order until reaching position ***(n + m) / 2***, then calculating the median value.
    c. Finally, it is possible to adapt binary search to reduce the complexity of your algorithm to sublinear complexity.