Lecture 1 - Arrays
💡 **Q1.** Given an array of size N. The task is to find the maximum and the minimum element of the array using the minimum number of comparisons.
Examples:
Input: arr[] = {3, 5, 4, 1, 9}
Output: Minimum element is: 1
Maximum element is: 9
TC : O(n)
SC: O(n)
Solution:
Java
Code Link :- https://pastebin.com/hMGGWh8h
****
💡 **Q2.** You are given an array prices where prices[i] is the price of a given stock on the ith day.You want to maximize your profit by choosing a single day to buy one stock and choosing a different day in the future to sell that stock.
Return the maximum profit you can achieve from this transaction. If you cannot achieve any profit, return 0.
Example :
Input: prices = [7,1,5,3,6,4]
Output: 5
Explanation: Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = 6-1 = 5.
Note that buying on day 2 and selling on day 1 is not allowed because you must buy before you sell.
Solution:
Java
Code link :- https://pastebin.com/BNS5dH6C
TC: O(n)
SC : O(1)
****
<aside>
💡 **Q3.** Given an integer array nums, find a subarray that has the largest product, and return the product.

The test cases are generated so that the answer will fit in a 32-bit integer.

Example:

Input: nums = [2,3,-2,4]

Output: 6

Explanation: [2,3] has the largest product 6.

Solution:

Java

Code link :- https://pastebin.com/r0xp1KhC

TC : O(n)
SC: O(1
</aside>
****
<aside>
💡 **Q4.** Given an integer array nums, return all the triplets [nums[i], nums[j], nums[k]] such that i != j, i != k, and j != k, and nums[i] + nums[j] + nums[k] == 0.

Notice that the solution set must not contain duplicate triplets.

Example:

Input: nums = [-1,0,1,2,-1,-4]

Output: [[-1,-1,2],[-1,0,1]]

Explanation:

nums[0] + nums[1] + nums[2] = (-1) + 0 + 1 = 0.

nums[1] + nums[2] + nums[4] = 0 + 1 + (-1) = 0.

nums[0] + nums[3] + nums[4] = (-1) + 2 + (-1) = 0.

The distinct triplets are [-1,0,1] and [-1,-1,2].

Notice that the order of the output and the order of the triplets does not matter.

Solution:

Java

Code link :- https://pastebin.com/GXbv3v9g
TC: O(n^2)
SC : O(n) depending on which sorting algo is used
</aside>

****

<aside>
💡 **Q5.** Given an integer array nums and an integer k, return the kth largest element in the array. Note that it is the kth largest element in the sorted order, not the kth distinct element.

Example 1:

Input: nums = [3,2,1,5,6,4], k = 2

Output: 5

Solution:

Java :

Code link :- https://pastebin.com/mr8TFWQP
TC : O(n log k)
TC : O(k)
</aside>
****
<aside>
💡 **Q6.** Given an integer array nums and an integer k, return the kth smallest element in the array. Note that it is the kth smallest element in the sorted order, not the kth distinct element.

Example 1:

Input: nums = [3,2,1,5,6,4], k = 2

Output: 2

JAVA

Code link: https://pastebin.com/drqYCbLQ

</aside>