

Lesson:



Problems based on sortings – 2



Pre-Requisites

- Arrays basics
- Sorting algorithms

List of Concepts Involved

- Problems based on sorting algorithms

Problem 1

Given an array where all its elements are sorted in increasing order except two swapped elements, sort it in linear time. Assume there are no duplicates in the array.

For example,

Input: $A[] = [3, 8, 6, 7, 5, 9, 10]$

Output: $A[] = [3, 5, 6, 7, 8, 9, 10]$

The idea is to start from the second array element and compare every element with its previous element. We take two pointers, x and y , to store the conflict's location. If the previous element is greater than the current element, update x to the previous element index and y to the current element index. If we find that the previous element is greater than the current element, update y to the current element index. Now that we have got the x and y indices, swap the elements at index x and y .

<https://pastebin.com/Ptt4WWcB>

```
Test x
/Library/Java/JavaVirtualMachines/jdk-19.jdk
3 5 6 7 8 9 10
Process finished with exit code 0
```

Problem 2: Given an array of size N containing only 0s, 1s, and 2s; sort the array in ascending order.

Input:

$N = 6$

$arr[] = \{0\ 2\ 1\ 2\ 0\ 0\}$

Output:

0 0 0 1 2 2

Approach:

- Keep three counters $count0$ to count 0s, $count1$ to count 1s, and $count2$ to count 2s
- Traverse through the array and increase the count of $count0$ if the element is 0, increase the count of $count1$ if the element is 1 and increase the count of $count2$ if the element is 2
- Now again traverse the array and replace the first $count0$ elements with 0, the next $count1$ elements with 1, and the next $count2$ elements with 2.

Code link

<https://pastebin.com/LitRaHXv>

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents
Enter the length of the array:
6
Enter the elements of the array:
0 2 1 2 0 0
0 0 0 1 2 2
Process finished with exit code 0
```

Problem 3: Given an array of positive and negative integers, segregate them in linear time and constant space. The output should print all negative numbers, followed by all positive numbers.

For example,

Input: [19, -20, 7, -4, -13, 11, -5, 3]

Output: [-20 -4 -13 -5 7 11 19 3]

- The idea is to use 0 as a pivot element and make one pass of the partition process. The resultant array will satisfy the given constraints.
- In partition function , each time we find a negative number, `pIndex` is incremented and that element would be placed before the pivot

This approach is demonstrated below in Java:

Code

<https://pastebin.com/VJXjn7Vp>

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/t
-20 -4 -13 -5 7 11 19 3
Process finished with exit code 0
```

Upcoming Class Teasers:

- Binary Search