Sorting [Day 12]

14 September 2024 20:02

Sorting [Day 12].

Problem Statement: Bubble Sort

You are required to implement the Bubble Sort algorithm on an array and print three values:

1. The number of swaps it took to sort the array.
2. The first element in the array after sorting.
3. The last element in the array after sorting.

Input Format:

- The first line contains an integer `n`, the size of the array.
- The second line contains `n` space-separated integers representing the array `a`.

Output Format:

- Print the required three values:
  1. Number of swaps it took to sort the array.
  2. First element in the array after sorting.
  3. Last element in the array after sorting.

Example 1:
Input:
3
1 4 2
Output:
1
1
4

Example 2:
Input:
5
5 4 3 2 1
Output:
10
1
5


```
1 to n
   1 to n - i - 1
   if(arr[j] > arr[j+1]){
      Swap;
      count++;
   }
```

swapCount = 10;

```
import java.util.Scanner;
public class bubbleSort {
    public static void bubblesort(int[]arr , int n){
        int swapcount = 0 ;
```

```java
public static void bubblesort(int[]arr , int n){
    int swapcount = 0 ;

    for(int i =0; i < n -1 ; i++){
        for(int j = 0; j < n-i-1; j++){
            if(arr[j] > arr[j+1]){
                int temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
                swapcount++;
            }
        }
    }
    System.out.println(swapcount);
    System.out.println(arr[0]);
    System.out.println(arr[n-1]);
}
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for(int i = 0 ; i< n; i++){
        arr[i] = scn.nextInt();
    }
    bubblesort(arr , n);
}
}
```

```
5
5
4
3
2
1
10
1
5
```

O(N^2)

```java
for(int i =0; i < n -1 ; i++){
    for(int j = 0; j < n-i-1; j++){
        if(arr[j] > arr[j+1]){
            int temp = arr[j];
            arr[j] = arr[j+1];
            arr[j+1] = temp;
            swapcount++;
        }
    }
}
```

1 | 2 3 4 6 8

swapCount = 12
arr[0] = 1
arr[n-1] = 8

Arrays.Sort() - O(NLogN)

Problem Statement: Odd-Even Sorting

You are given an array of integers, and your task is to sort the array in ascending order using the **Odd-Even sort** algorithm.

Input Format:

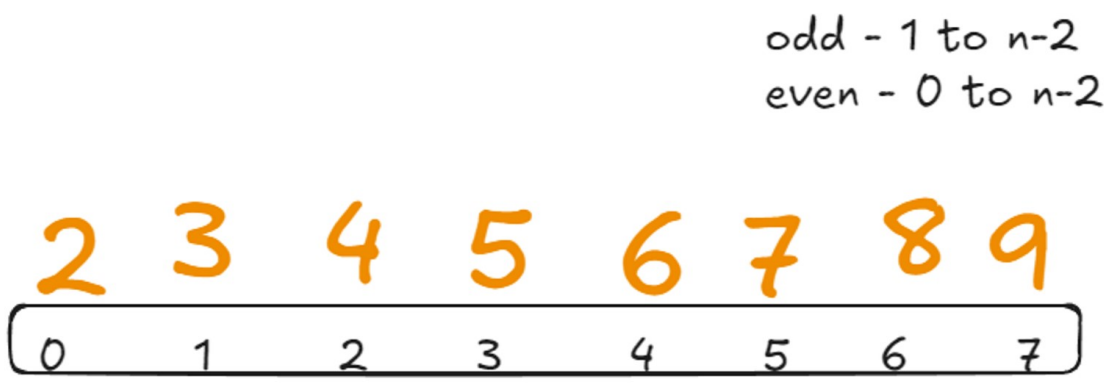
- The first line contains an integer `N`, representing the number of elements in the array.
- The second line contains `N` space-separated integers representing the elements of the array.

Output Format:

- Print the sorted array in ascending order using the Odd-Even sort.

Example 1:
Input:
8
9 8 7 6 5 4 3 2
Output:
2 3 4 5 6 7 8 9

Example 2:
Input:
5
10 15 12 17 14
Output:
10 12 14 15 17




```
import java.util.Scanner;
```

```java
import java.util.Scanner;
public class OddEvenSort {
    public static void oddEvenSort(int[] arr , int n){
        boolean isSorted = false;

        while(!isSorted){
            isSorted = true;
            // Odd index
            for(int i = 1; i<=n-2; i+=2){
                if(arr[i] > arr[i+1]){
                    int temp = arr[i];
                    arr[i] = arr[i+1];
                    arr[i+1] = temp;
                    isSorted = false;
                }
            }
            // Even index
            for(int i = 0; i<=n-2; i+=2){
                if(arr[i] > arr[i+1]){
                    int temp = arr[i];
                    arr[i] = arr[i+1];
                    arr[i+1] = temp;
                    isSorted = false;
                }
            }
        }
    }
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();

        int[] arr = new int[n];
        for(int i = 0 ; i<n ; i++){
            arr[i] = scn.nextInt();
        }

        oddEvenSort(arr , n);
        for(int i=0 ; i<n ; i++){
            System.out.print(arr[i] + " ");
        }
    }
}
```

```
7
9
8
7
6
5
4
3
3 4 5 6 7 8 9
```

---

1. Find the Median

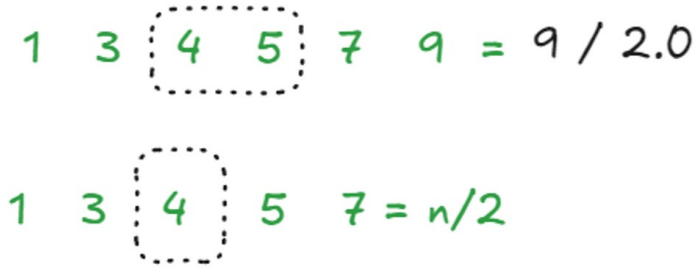
1. Find the Median

Problem Statement:

Given an unsorted array of integers, find the median of the array. The median is the middle value of the sorted array. If the number of elements is odd, it's the middle element. If it's even, the median is the average of the two middle elements.

```
Input: [7, 1, 3, 4, 5, 9]
Output: 4.5

Input: [7, 1, 3, 4, 5]
Output: 4
```




```java
import java.util.Arrays;
public class MedianFinder {
    public static double findMedian(int[] arr){
        Arrays.sort(arr);
        int n = arr.length;
        if(n % 2 !=0){
            return arr[n/2];
        }else{
            return (arr[n/2] + arr[(n/2)-1]) / 2.0;
        }
    }
    public static void main(String[] args) {
        int[] arr = {7,1,3,4,5,9};
        System.out.println("Median : " + findMedian(arr));
    }
}
```

```
MedianFinder
Median : 4.5
```

4. 0-1 Sorting

Problem Statement:

Given an array consisting only of 0s and 1s, sort the array in linear time.

Given an array consisting only of 0s and 1s, sort the array in linear time.

```
Input: [0, 1, 1, 0, 1, 0]
Output: [0, 0, 0, 1, 1, 1]
```


```
import java.util.Arrays;
public class Sort01 {
    public static void sort01(int[] arr){
        int left = 0;
        int right = arr.length - 1;
        while(left < right){
            if(arr[left] == 1 && arr[right] == 0){
                int temp = arr[left];
                arr[left] = arr[right];
                arr[right] = temp;
            }
            if(arr[left] == 0){
                left++;
            }
            if(arr[right] == 1){
                right--;
            }
        }
    }
    public static void main(String[] args) {
        int[] arr = {0,1,1,0,1,0};
        sort01(arr);
        System.out.println(Arrays.toString(arr));
    }
}
```

Tips for Resume & LinkedIn

1. Resume should be up to date.
2. Resume should be shortlisted based on ATS.

2. Resume should be shortlisted based on ATS.
3. Create at 3-4 different type of resume.
4. Make Sure , you are providing good details about the projects , Achievements , etc.
5. Discussed different ways of looking for job in LinkedIn.
6. Always try to provide good context of connection. Why you reached to them, What opportunities you are looking for, And how that connection will help , Drop the Resume as well.

Completed

Attendance Code: CFCAA5C1