https://course.acciojob.com/idle?question=98f32292-cfdb-45de-8d1d-1597b161e0d9

- MEDIUM
- Max Score: 40 Points

Next Permutation

Implement the next permutation, which rearranges numbers into the numerically next greater permutation of numbers for a given array A of size N.

If such arrangement is not possible, it must be rearranged as the lowest possible order i.e., sorted in an ascending order.

Input Format

First line contains integer `N`

Second line contains `N` integers `arr[i]`.

Output Format

Print an array of integers, representing the next permutation of the given array.

Example 1

Input

```
3
1 2 3
```

Output

```
1 3 2
```

Explanation

132 is the next permutation of 123

Example 2

Input

```
3
3 2 1
```

Output

```
1 2 3
```

Explanation

Since the number is in its maximum form of digits, we print the sorted order, i.e 1, 2, 3

Constraints

```
1 <= N <= 10^5
```

```
0 <= arr[i] <= 10^9
```

Topic Tags

- **Arrays**

My code

```
// n javaimport java.util.*;
import java.lang.*;
import java.io.*;

public class Main {

    public static int[] nextPermutation(int[] arr) {
            int n=arr.length;
       //Write your code here
            int pivot=-1;
            for(int i=n-1;i>0;i--)
                {
                    if(arr[i]>arr[i-1])
                    {
                        pivot=i-1;
                        break;
                    }
                }
// if end is given so need start so sort
            if(pivot<0)
            {
                Arrays.sort(arr);
               return arr;
            }

            //now find smallest of grater of pivot element index
```

```
            int j;
            for (j = n - 1; j > pivot; j--) {
          if (arr[j] > arr[ pivot]) {
            break;
          } }
 // Swap the pivot and successor
       int temp = arr[pivot];
       arr[pivot] = arr[j];
        arr[j] = temp;
            //now pivot+1 karo then sort the end of this arry from j to
last
                    j=pivot+1;
                    int k=n-1;
                    while(j<k)//swap
                          {
                           temp = arr[k];
             arr[k] = arr[j];
              arr[j] = temp;
                                k--;
                                j++;
                          }
            return arr;
   }
  public static void main (String[] args)
       {
            Scanner sc = new Scanner(System.in);
          int N = sc.nextInt();
       int arr[] =new int[N];
       for(int i=0;i<N;i++)
       {
```

```
            arr[i]=sc.nextInt();
        }
        int result[] = nextPermutation(arr);
        for(int i=0;i<N;i++)
        {
            System.out.print(result[i]+" ");
        }
        }
}
```