**Smallest number on left**

Link:- [Smallest number on left | Practice | GeeksforGeeks](https://practice.geeksforgeeks.org/problems/smallest-number-on-left3403/1?utm_source=gfg&utm_medium=article&utm_campaign=bottom_sticky_on_article)

Given an array **a**of integers of length **n**, find the nearest smaller number for every element such that the smaller element is on left side.If no small element present on the left print -1.

**Example 1:**

**Input:** n = 3

a = {1, 6, 2}

**Output:** -1 1 1

**Explaination:** There is no number at the

left of 1. Smaller number than 6 and 2 is 1.

**Example 2:**

**Input:** n = 6

a = {1, 5, 0, 3, 4, 5}

**Output:** -1 1 -1 0 3 4

**Explaination:** Upto 3 it is easy to see

the smaller numbers. But for 4 the smaller

numbers are 1, 0 and 3. But among them 3

is closest. Similary for 5 it is 4.

//User function Template for Java

class Solution{

static List<Integer> leftSmaller(int n, int a[])

{

//code here

ArrayList<Integer> solution=new ArrayList<>();

solution.add(-1);

Stack<Integer> st=new Stack<>();

st.push(a[0]);

for(int i=1;i<n;i++){

if(st.peek()<a[i]){

solution.add(st.peek());

st.push(a[i]);

}

else{

while(!st.isEmpty()){

if(st.peek()<a[i]){

solution.add(st.peek());

st.push(a[i]);

break;

}

else{

st.pop();

}

}

if(st.isEmpty()){

solution.add(-1);

st.push(a[i]);

}

}

}

return solution;

}

}