6

Verify Preorder Sequence in Binary Search Tree

Difficulty: MEDIUM

Avg. time to solve
30 min

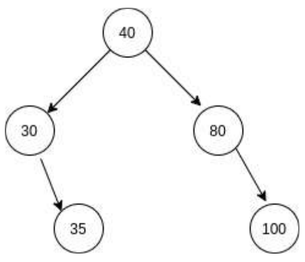
Success Rate
70%

Contributed By
Sounak Majumder | Level 1

Problem Statement Suggest Edit

You are given an array/list ‘ARR’ consisting of ‘N’ distinct integers. You need to check whether ‘ARR’ can represent the Preorder Traversal of a Binary Search Tree.

You should return true if ‘ARR’ can represent Preorder Traversal of a Binary Search Tree, otherwise return false.



Example:

Consider ‘ARR’ = [40, 30, 35, 80, 100]. You can see that it is the preorder traversal of the Binary Search Tree shown above.

Thus, we should return true in this case.

Input Format:

The first line of input contains an integer ‘T’ denoting the number of test cases. then ‘T’ test cases follow.

The first line of each test case consists of a single integer ‘N’ representing the size of the list/array ‘ARR’.

The second line of each test case consists of ‘N’ single space-separated integers representing list/ array ‘ARR’.

Output Format :

For each test case, print a single line containing a string ‘True’ if ‘ARR’ can represent Preorder Traversal of a Binary Search Tree, otherwise print ‘False’.

The output of each test case will be printed in a separate line.

Note:

You do not need to print anything, it has already been taken care of. Just implement the given function.

Constraints:

1 <= T <= 50
1 <= N <= 10 ^ 4
1 <= ARR[i] <= 10 ^ 9

Where ‘ARR[i]’ is the element of the given array ‘ARR’ at index ‘i’.

Time limit: 1 sec.

Sample Input 1:

```
2
1
1
5
40 30 35 80 100
```

Sample Output 1:

```
True
True
```

Explanation Of Sample Input 1:

Test case 1:
There is only one element is ‘ARR’, and it is the preorder traversal of a Binary Search tree having a single node with value 1.

Test case 2:
See the problem statement for an explanation.

Sample Input 2:

```
2
3
2 4 1
6
5 2 3 1 7 8
```

Sample Output 2:

```
False
False
```

C++ (g++ 5.4)

```cpp
#include <bits/stdc++.h>
bool isBSTPreorder(vector<int> &arr)
{
    // Write your code here.
    stack<int>st;
    int root=0;
    for(int i=0;i<arr.size();i++){
        if(arr[i]<root){
            return false;
        }
        while(!st.empty() and st.top()<arr[i]){
            root=st.top();
            st.pop();
        }
        st.push(arr[i]);
    }
    return true;
}
```

Console

Sample Test Case Custom Test Case

Download Test Cases

Test Case 1
Correct Answer

Test Case 2
Correct Answer

Test Case 3
Correct Answer

Test Case 4
Correct Answer

Test Case 5
Correct Answer

Test Case Input

2
1
1
5
40 30 35 80 100

Your Output

True
True

Desired Output

True
True

Note: Correct space Incorrect space

Previous Next Show Hint Last saved on 5:52:12 AM Run Code Submit