

## Trees and Graphs

Raju wasn't able to distinguish the difference between a binary tree and binary search tree. He went to his teacher to know the concepts. Later Raju decided to frame a solution that finds whether the given binary tree is a binary search tree or not. Help Raju to find a solution.

### Input:

First line, denotes the number of nodes in a binary tree.

Second line, denotes nodes of a binary tree.

### Output:

Print "Is BST" or "Not a BST".

### Solution:

```
class Node:
    def __init__(self,data):
        self.data=data
        self.left=None
        self.right=None
class BinaryTree:
    def __init__(self):
        self.root=None
def createBinaryTree(lst,n,node,i):
    if i<n:
        if lst[i]==-1:
            return node
        temp=Node(lst[i])
        node=temp
        node.left=createBinaryTree(lst,n,node.left,(2*i)+1)
        node.right=createBinaryTree(lst,n,node.right,(2*i)+2)
    return node
def isBST(root,l=None,r=None):
    if root==None:
        return True
    if l!=None and root.data<=l.data:
        return False
    if r!=None and root.data>=r.data:
        return False
    return isBST(root.left,l,root) and isBST(root.right,root,r)
n=int(input())
lst=[int(x) for x in input().split()]
```

```
tree=BinaryTree()
tree.root=createBinaryTree(lst,n,tree.root,0)
if isBST(tree.root,None,None):
    print("Is BST")
else:
    print("Not a BST")
```

**Test Case1:**

**Input:**

5  
4 2 5 1 3

**Output:**

Is BST

**Test Case2:**

**Input:**

5  
5 2 4 1 3

**Output:**

Not a BST

**Test Case3:**

**Input:**

7  
2 3 14 5 7 6

**Output:**

Not a BST

**Test Case 4:**

**Input:**

4  
4 2 5 1

**Output:**

Is BST