Valid BST

by abhiranjan

Problem Submissions Leaderboard Discussions

A binary tree is a tree where each node has at most two children. It is characterized by any of the following properties:

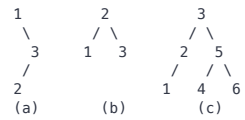
1. It can be an empty tree, where root = null.
2. It can contain a root node which contain some value and two subtree, left subtree and right subtree, which are also binary tree.

A binary tree is a binary search tree (BST) if all the non-empty nodes follows both two properties:

1. Each node's left subtree contains only values less than it, and
2. Each node's right subtree contains only values greater than it.

Preorder traversal is a tree traversal method where the current node is visited first, then the left subtree and then the right subtree. More specifically, let's represent the preorder traversal of a tree by a list. Then this list is constructed in following way:

1. If the tree is empty, then this list be a null list.
2. For non-empty tree, let's represent the preorder of left subtree as L and of right subtree as R. Then the preorder of tree is obtained by appending L to current node, and then appending R to it.



For the above trees, preorder will be

```
(a) 1 3 2
(b) 2 1 3
(c) 3 2 1 5 4 6
```

Given a list of numbers, determine whether it can represent the preorder traversal of a binary search tree(BST).

Input
The first line contains the number of test cases, T. Then T test cases follow. The first line of each test case contains the number of nodes in the tree, N. In next line there will a list of N unique numbers, where each number is from set [1, N].

Output
For each test case, print "YES" if there's exist a BST whose preorder is equal to the list otherwise "NO" (without quotes).

Constraints
1 <= T <= 10
1 <= N <= 100

Sample Input

```
5
3
1 2 3
3
2 1 3
6
3 2 1 5 4 6
4
1 3 4 2
5
3 4 5 1 2
```

Sample Output

```
YES
YES
```

```
YES
NO
NO
```

Explanation

First three cases are from examples. And last two test cases are invalid because the subtree for 3 is not valid as 2 and 4 are in the wrong order.

Submissions: 661
Max Score: 20
Difficulty: Medium

Rate This Challenge:

More

If you cannot find your favorite language here, please check out the algorithm domain. We support over 40 popular languages.

Current Buffer (saved locally, editable) Racket

```
#lang racket
; Enter your code here. Read input from STDIN. Print output to STDOUT
(define (any-list f l)
  (cond ((eq? l '()) #f)
        ((f (car l)) #t)
        (else (any-list f (cdr l)))))

(define (take-while f l)
  (cond ((eq? l '()) '())
        ((f (car l)) (cons (car l) (take-while f (cdr l))))
        (else (take-while f (cdr l)))))

(define (drop-while f l)
  (cond ((eq? l '()) '())
        ((f (car l)) (drop-while f (cdr l)))
        (else l)))

(define (bst-from-preorder? l)
  (cond ((eq? l '()) #t)
        (else
         (letrec ((list-left (take-while (lambda (x) (< x (car l))) (cdr l)))
                  (list-right (drop-while (lambda (x) (< x (car l))) (cdr l))))
           (cond
             ((any-list (lambda (x) (> x (car l))) list-left) #f)
             ((any-list (lambda (x) (< x (car l))) list-right) #f)
             (else (and (bst-from-preorder? list-left)
                        (bst-from-preorder? list-right))))))))

(define (eval-test n l)
  (if (= n 0) l
      (eval-test (- n 1) (append l (list (read))))))

(define (go tests)
  (if (= tests 0) #f
      (begin
        (displayln (if (bst-from-preorder? (eval-test (read) '())) "YES" "NO"))
        (go (- tests 1)))))

(define tests (read))
(define repl-test (go tests))
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code