Problem 255: Verify Preorder Sequence in Binary Search Tree

Problem Information

Difficulty: Medium
Acceptance Rate: 51.53%
Paid Only: Yes
Tags: Array, Stack, Tree, Binary Search Tree, Recursion, Monotonic Stack, Binary Tree

Problem Description

Given an array of **unique** integers `preorder`, return `true` _if it is the correct preorder traversal sequence of a binary search tree_.

Example 1:

![](https://assets.leetcode.com/uploads/2021/03/12/preorder-tree.jpg)

Input: preorder = [5,2,1,3,6] **Output:** true

Example 2:

Input: preorder = [5,2,6,1,3] **Output:** false

Constraints:

* `1 <= preorder.length <= 104` * `1 <= preorder[i] <= 104` * All the elements of `preorder` are **unique**.

Follow up: Could you do it using only constant space complexity?

Code Snippets

C++:

```
class Solution {
public:
bool verifyPreorder(vector<int>& preorder) {

}
};
```

Java:

```
class Solution {
public boolean verifyPreorder(int[] preorder) {

}
}
```

Python3:

```
class Solution:
def verifyPreorder(self, preorder: List[int]) -> bool:
```