173. Binary Search Tree Iterator

题目描述：https://leetcode.com/problems/binary-search-tree-iterator/

> 给定一个二叉查找树，设计一种数据结构似的可以用hasnext()，next(), 按照大小依次返回结果。

解题思路：

其实就是求前序遍历

代码：

```
/**
 * Definition for binary tree
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class BSTIterator {
public:
    vector<TreeNode*> preorder;
    int i;
    BSTIterator(TreeNode *root) {
        i = 0;
        findPreorder(root);
    }
    void findPreorder(TreeNode* root) {
        if(root == NULL) {
            return;
        }
        if(root->left == NULL) {
            preorder.push_back(root);
        }
        else {
            findPreorder(root->left);
            preorder.push_back(root);
        }
        findPreorder(root->right);
    }

    /** @return whether we have a next smallest number */
    bool hasNext() {
        return i < preorder.size();
    }

    /** @return the next smallest number */
    int next() {
        return preorder[i++]->val;
    }
};

/**
 * Your BSTIterator will be called like this:
 * BSTIterator i = BSTIterator(root);
 * while (i.hasNext()) cout << i.next();
 */
```