144. Binary Tree Preorder Traversal

Description | Hints | Submissions | Discuss | Solution

Pick One

Given a binary tree, return the *preorder* traversal of its nodes' values.

Example:

```
Input: [1,null,2,3]
   1
    \
     2
    /
   3

Output: [1,2,3]
```

Follow up: Recursive solution is trivial, could you do it iteratively?

```
public class L144 {

    public class TreeNode {
        int val;
        TreeNode left;
        TreeNode right;
        public TreeNode(int x) {
            val = x;
        }
    }
    /*
     * 这是树的先序遍历，先序遍历的规则是：对于遇到的每一个节点，先访问节点本身，然后是左子树根节点。
     */
      public List<Integer> preorderTraversal(TreeNode root) {
          List<Integer> result = new LinkedList<Integer>();
          if(root == null)
              return result;
          LinkedList<TreeNode> stack = new LinkedList<TreeNode>();
          stack.push(root);
          while (!stack.isEmpty()) {
            TreeNode top = stack.pop();
            /*
             * 因为栈是先进后出，所以先插入右儿子节点。
             */
            if(top != null){
                result.add(top.val);
                stack.push(top.right);
                stack.push(top.left);
            }
        }
          return result;
      }
}
```