1. Binary Tree Preorder Traversal

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?

**解** 使用栈

/\*  
功能：先序遍历非递归算法  
描述：  
（1）不空则访问、进栈、往左走  
（2）空则出栈、往右走  
\*/

class Solution {  
public:  
 vector<int> preorderTraversal(TreeNode\* root) {  
 stack<TreeNode\*>s;  
 vector<int>path;  
 TreeNode \*p = root;  
 while(p || !s.empty()){  
 while(p){  
 path.push\_back(p->val);  
 s.push(p);  
 p = p->left;  
 }  
 if(!s.empty()){  
 p = s.top();  
 s.pop();  
 p = p->right;  
 }  
 }  
 return path;  
 }  
};