# 题目

输入两个整数序列，第一个序列表示栈的压入顺序，请判断第二个序列是否为该栈的弹出顺序。假设压入栈的所有数字均不相等。例如，序列 {1,2,3,4,5} 是某栈的压栈序列，序列 {4,5,3,2,1} 是该压栈序列对应的一个弹出序列，但 {4,3,5,1,2} 就不可能是该压栈序列的弹出序列。

**示例 1：**

输入：pushed = [1,2,3,4,5], popped = [4,5,3,2,1]

输出：true

解释：我们可以按以下顺序执行：

push(1), push(2), push(3), push(4), pop() -> 4,

push(5), pop() -> 5, pop() -> 3, pop() -> 2, pop() -> 1

**示例 2：**

输入：pushed = [1,2,3,4,5], popped = [4,3,5,1,2]

输出：false

解释：1 不能在 2 之前弹出。

**提示：**

0 <= pushed.length == popped.length <= 1000

0 <= pushed[i], popped[i] < 1000

pushed 是 popped 的排列。

**注意：**本题与主站 946 题相同：

https://leetcode-cn.com/problems/validate-stack-sequences/

**类似题目：**Leetcode 946

# 分析

## 方法一：贪心算法

## 方法二：栈

**思路：**

**代码：**

bool validateStackSequences(vector<int>& pushed, vector<int>& popped) {

stack<int> s;

int n = pushed.size();

for(int i = 0,j=0; i < n; i++){//模拟进栈

s.push(pushed[i]);//先直接进栈

//然后根据出栈序列决定是否出栈

while(!s.empty()&&j<n&&s.top()==popped[j]) s.pop(),j++;//出栈

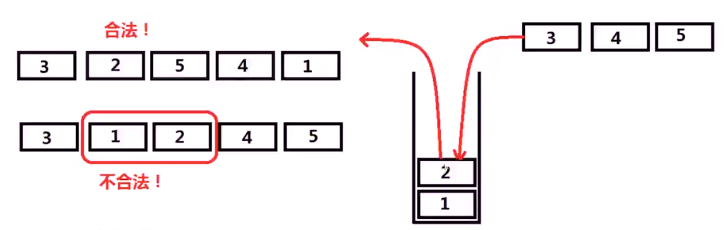
}

return s.empty() ? true:false;//若合法，则此时栈一定是空的

}

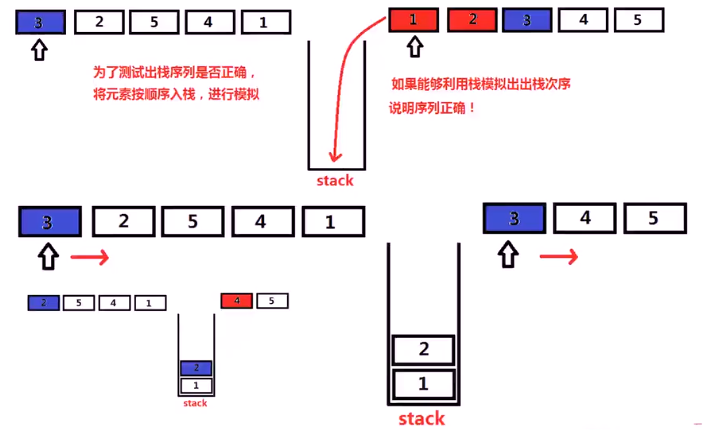
# 拓展

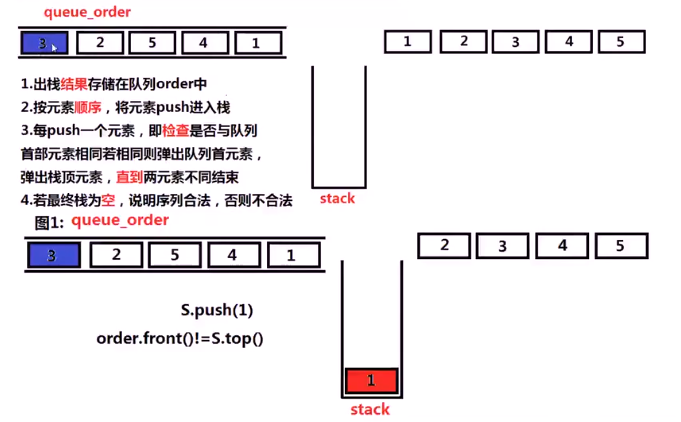
**题目：**已知从1至n的数字序列，按顺序入栈，每个数字入栈后即可出栈，也可在栈中停留，等待后面的数字入栈出栈后，该数字再出栈，求该数字序列的出栈序列是否合法？

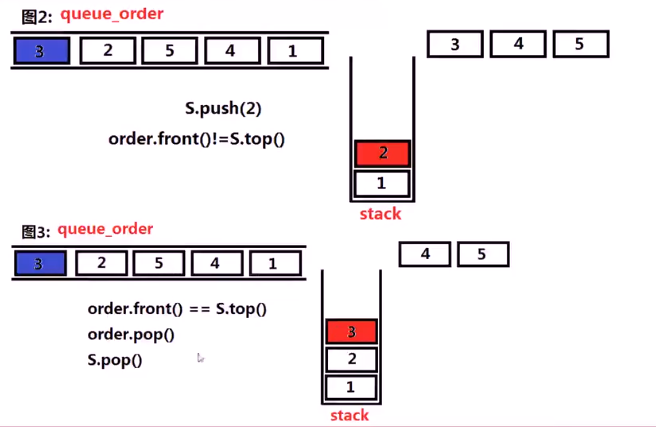


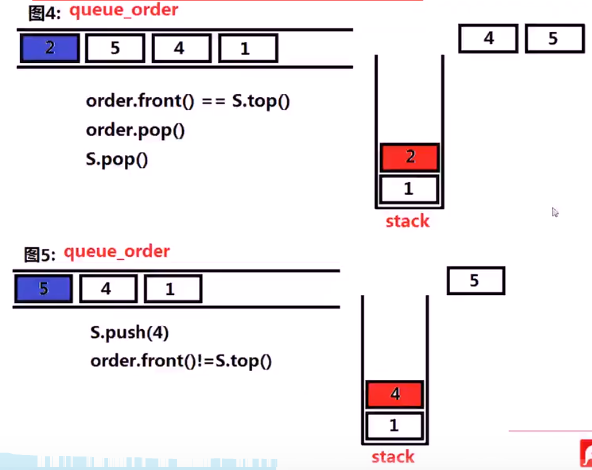
注：poj 1363 Rail

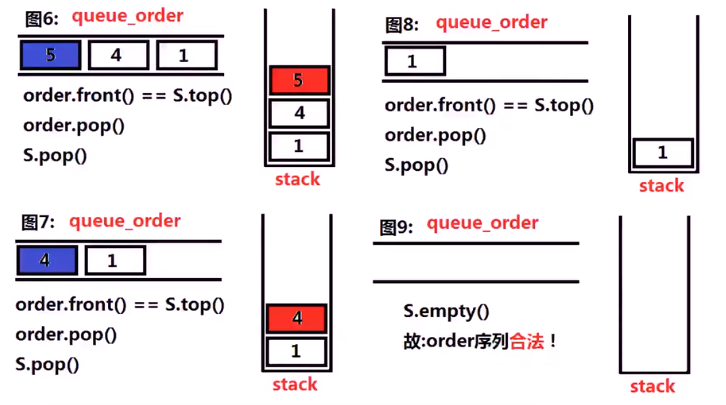
**分析：**



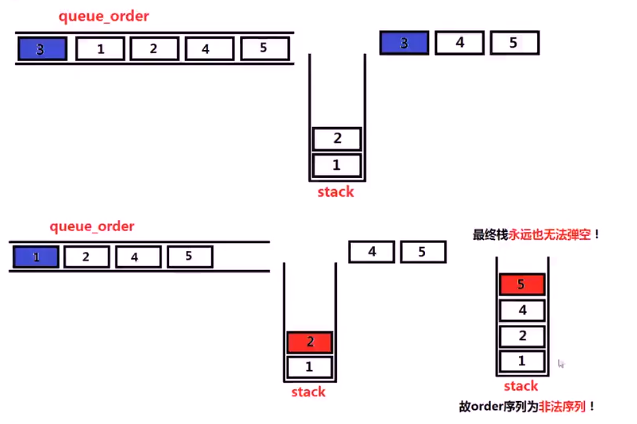




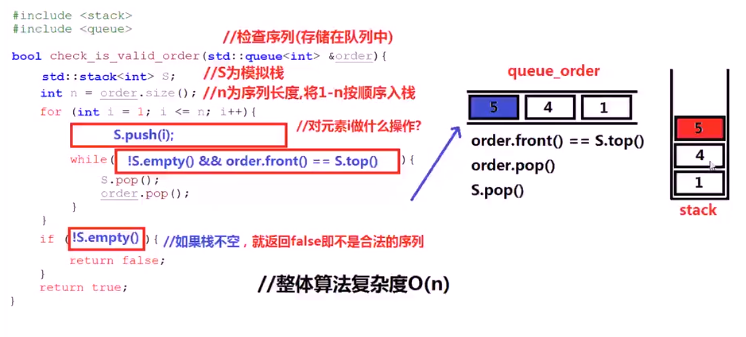




不合法：



**代码：**



**测试：**

