# 题目

给定 pushed 和 popped 两个序列，每个序列中的值都不重复，只有当它们可能是在最初空栈上进行的推入 push 和弹出 pop 操作序列的结果时，返回 true；否则，返回 false 。

**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。

# 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

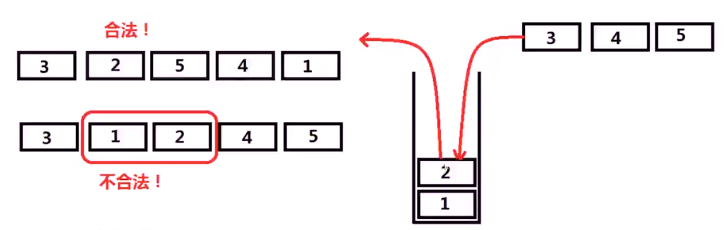
}

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

}

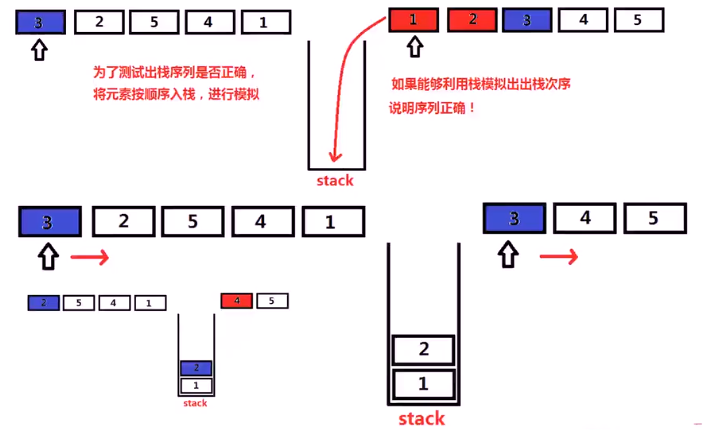
# 拓展

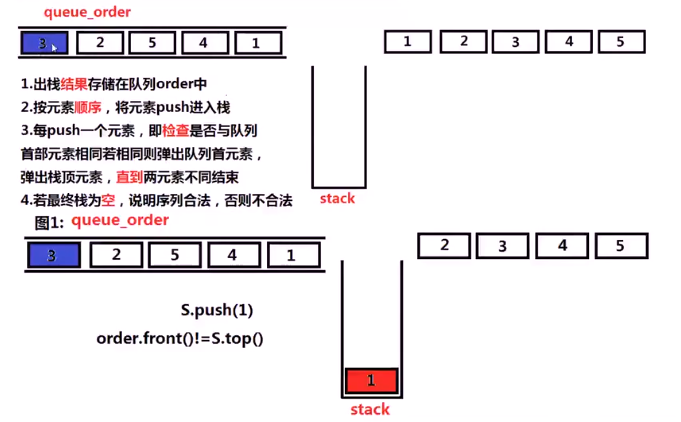
**题目：**已知从1至n的数字序列，按顺序入栈，每个数字入栈后即可出栈，也可在栈中停留，等待后面的数字入栈出栈后，该数字再出栈，求该数字序列的出栈序列是否合法？

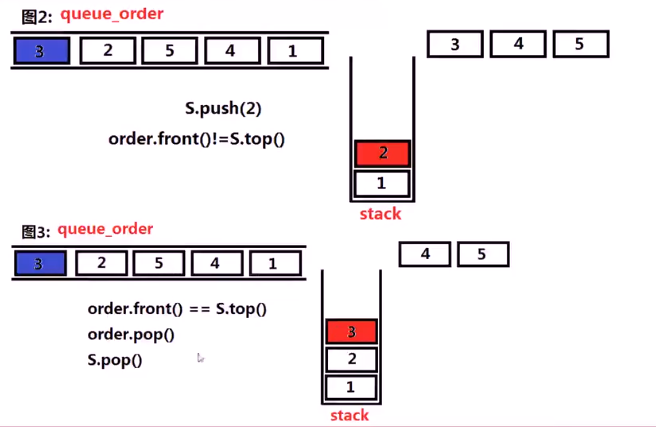


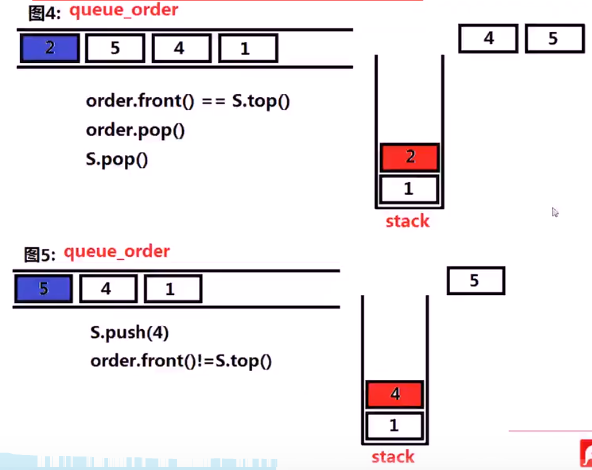
注：poj 1363 Rail

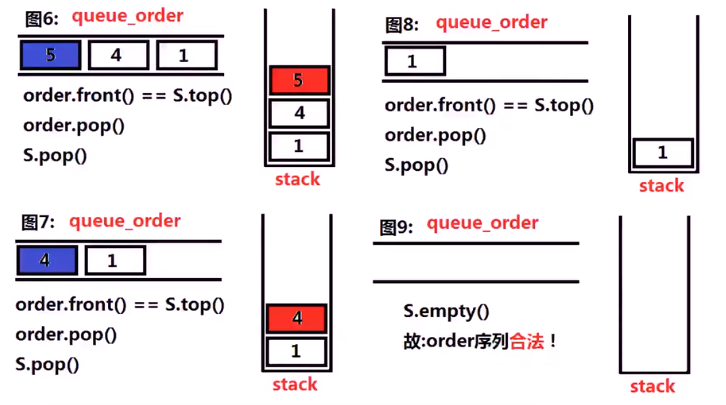
**分析：**



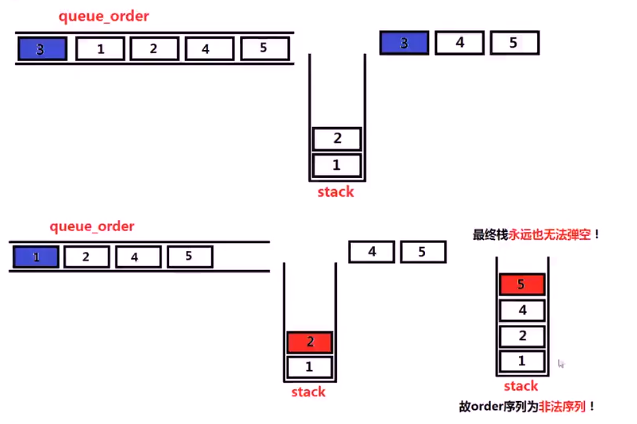




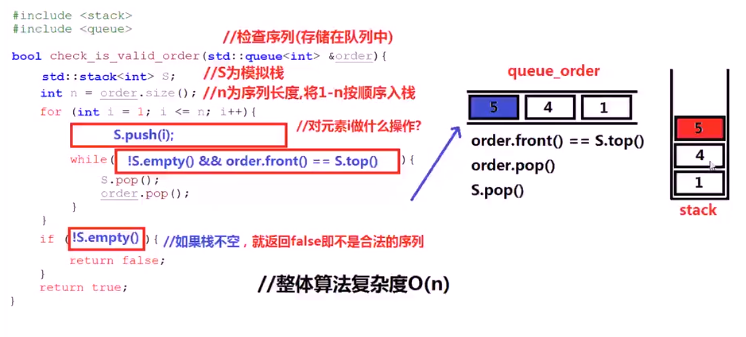




不合法：



**代码：**



**测试：**

