# 题目

读者来到图书馆排队借还书，图书管理员使用两个书车来完成整理借还书的任务。书车中的书从下往上叠加存放，图书管理员每次只能拿取书车顶部的书。排队的读者会有两种操作：

push(bookID)：把借阅的书籍还到图书馆。

pop()：从图书馆中借出书籍。

为了保持图书的顺序，图书管理员每次取出供读者借阅的书籍是 最早 归还到图书馆的书籍。你需要返回 每次读者借出书的值 。

如果没有归还的书可以取出，返回 -1 。

示例 1：

输入：

["BookQueue", "push", "push", "pop"]

[[], [1], [2], []]

输出：[null,null,null,1]

解释：

MyQueue myQueue = new MyQueue();

myQueue.push(1); // queue is: [1]

myQueue.push(2); // queue is: [1, 2] (leftmost is front of the queue)

myQueue.pop(); // return 1, queue is [2]

提示：

1 <= bookID <= 10000

最多会对 push、pop 进行 10000 次调用

# 分析

## 方法一：队列

class CQueue {

private:

stack<int> \_in;

stack<int> \_out;

public:

CQueue() {

}

void appendTail(int value) {

\_in.push(value);

}

int deleteHead() {

if(\_in.empty()) return -1;

while(!\_in.empty())

{

\_out.push(\_in.top());

\_in.pop();

}

int ret = \_out.top();

\_out.pop();

while(!\_out.empty())

{

\_in.push(\_out.top());

\_out.pop();

}

return ret;

}

};

/\*\*

\* Your CQueue object will be instantiated and called as such:

\* CQueue\* obj = new CQueue();

\* obj->appendTail(value);

\* int param\_2 = obj->deleteHead();

\*/