**难度: 简单**

使用队列实现栈的下列操作：

push(x) -- 元素 x 入栈

pop() -- 移除栈顶元素

top() -- 获取栈顶元素

empty() -- 返回栈是否为空

注意:

你只能使用队列的基本操作-- 也就是 push to back, peek/pop from front, size, 和 is empty 这些操作是合法的。

你所使用的语言也许不支持队列。 你可以使用 list 或者 deque（双端队列）来模拟一个队列 , 只要是标准的队列操作即可。

你可以假设所有操作都是有效的（例如, 对一个空的栈不会调用 pop 或者 top 操作）。

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/implement-stack-using-queues

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

**思路(用的Deque来模拟队列):**

1). 创建Deque,然后进行push的时候,直接添加

2). pop的时候,直接pop它的最后一个(因为队列是先进先出,我们是先进后出)

|  |
| --- |
| public class MyStack {  private Deque<Integer> queue;  private int i = 0;  /\*\* Initialize your data structure here. \*/  public MyStack() {  queue = new ArrayDeque<>();  }  /\*\* Push element x onto stack. \*/  public void push(int x) {  queue.add(x);  i++;  }  /\*\* Removes the element on top of the stack and returns that element. \*/  public int pop() {  return queue.pollLast();  }  /\*\* Get the top element. \*/  public int top() {  return queue.peekLast();  }  /\*\* Returns whether the stack is empty. \*/  public boolean empty() {  return queue.isEmpty();  }  } |