

```

import java.util.Queue;
import java.util.concurrent.PriorityBlockingQueue;

/*
 * 两个队列实现一个栈
 *
 * 将queue1用作进栈出栈，queue2作为一个中转站
 *
 * 入栈时，直接压入queue1中
 *
 * 出栈时，先将queue1中的元素除最后一个元素外依次出队列，并压入队列queue2中，将留在queue1中的最后一个元素出队列即为出栈元素，最后还要把queue2中的元素再次
 */
public class StackByTwoQueue {

    Queue<Integer> queue1 = new PriorityBlockingQueue<>();
    Queue<Integer> queue2 = new PriorityBlockingQueue<>();

    public void push(int m) {
        queue1.add(m);
    }

    public int pop() {
        for(int i = 0; i < queue1.size() - 1; i++) {
            queue2.add(queue1.poll());
        }
        int m = queue1.poll();

        for(int i = 0; i < queue2.size(); i++) {
            queue1.add(queue2.poll());
        }

        return m;
    }
}

```