

剑指 Offer 59 - II. 队列的最大值

Saturday, June 18, 2022 1:13 PM

<https://leetcode.cn/problems/dui-lie-de-zui-da-zhi-lcof/>

中等

请定义一个队列并实现函数 **max_value** 得到队列里的最大值，
要求函数max_value、push_back 和 pop_front 的均摊时间复杂度都是 $O(1)$ 。
若队列为空，pop_front 和 max_value 需要返回 -1

示例 1:

输入:

```
["MaxQueue","push_back","push_back","max_value","pop_front","max_value"]  
[[],[1],[2],[0],[0],[0]]
```

输出: [null,null,null,2,1,2]

思路:

定义两个队列，一个正常队列 存数，一个双端队列 存单调递减值。（双端队列存的就是当前队列中的最大值，每次入新值：小的等的直接加、大的就说明要更新当前最大值，就往前删小的）

入队：while判断双端队列

非空：此时值大于双端队列末尾值，删出该末尾值，再循环判断
while结束 或 双端队列队列为空
此时值直接入两个队。

出队:

正常队列空则返回-1，非空 则队首出队并返回，
同时判断其是否等于双端队列队首值，
等则同时删除双端队列值，不等则只删除正常队列队首值返回。

最大值:

直接返回双端队列队首值即可。

例: 32123

普通队列	双端队列入	删除
3	3	
32	32	此时删: queue: 2 deque: 2
321	321	
3212	322	
32123	33	

code:

```
class MaxQueue {  
    Queue<Integer> q;
```

```

Deque<Integer> d;

public MaxQueue() {
    q = new LinkedList<Integer>();
    d = new LinkedList<Integer>();
}

public int max_value() {
    if (d.isEmpty()) {
        return -1;
    }
    return d.peekFirst();
}

public void push_back(int value) {
    // 双端非空 且 屁股小于当前 新值，往前循环删；屁股大与等于则不动
    // 意思是更新当前正常队列的最大值
    while (!d.isEmpty() && d.peekLast() < value) {
        d.pollLast();
    }
    // 正常入队
    d.offerLast(value);
    q.offer(value);
}

public int pop_front() {
    if (q.isEmpty()) {
        return -1;
    }
    // 删的时候注意，因为双端队列队首保存的是当前队列内的最大值，若此时删除的正好是最大值，就要同步删除。
    int ans = q.poll();
    if (ans == d.peekFirst()) {
        d.pollFirst();
    }
    return ans;
}
}

```