**难度: 简单**

写一个 RecentCounter 类来计算特定时间范围内最近的请求。

请你实现 RecentCounter 类：

RecentCounter() 初始化计数器，请求数为 0 。

int ping(int t) 在时间 t 添加一个新请求，其中 t 表示以毫秒为单位的某个时间，并返回过去 3000 毫秒内发生的所有请求数（包括新请求）。确切地说，返回在 [t-3000, t] 内发生的请求数。

保证 每次对 ping 的调用都使用比之前更大的 t 值。

示例：

输入：

["RecentCounter", "ping", "ping", "ping", "ping"]

[[], [1], [100], [3001], [3002]]

输出：

[null, 1, 2, 3, 3]

解释：

RecentCounter recentCounter = new RecentCounter();

recentCounter.ping(1); // requests = [1]，范围是 [-2999,1]，返回 1

recentCounter.ping(100); // requests = [1, 100]，范围是 [-2900,100]，返回 2

recentCounter.ping(3001); // requests = [1, 100, 3001]，范围是 [1,3001]，返回 3

recentCounter.ping(3002); // requests = [1, 100, 3001, 3002]，范围是 [2,3002]，返回 3

提示：

1 <= t <= 109

保证每次对 ping 调用所使用的 t 值都 严格递增

至多调用 ping 方法 104 次

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/number-of-recent-calls

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(使用队列):**

1). 先判断 传入的t-3000是否小于0,.如果小于0直接入队

2). 如果不小于0,那么就循环判断队列的值是否小于传入的t-3000,然后成立就一直出对队

3). 循环结束后最后就入队

4). 最后返回队中的个数

|  |
| --- |
| class RecentCounter {  private Queue<Integer> queue ;  public RecentCounter() {  queue = new ArrayDeque<>();  }  public int ping(int t) {  int res = t -3000;  if (res< 0) {  queue.add(t);  }else {  while (!queue.isEmpty() && queue.peek() < res) {  queue.remove();  }  queue.add(t);  }  return queue.size();  }  } |