

最近的请求次数

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
```

```
class RecentCounter {
public:
    //使用队列，不断检查队列头部元素和当前插入的时间之差是否大于 3000，如果大于就
    弹出来头部元素。
    queue<int> q;
    RecentCounter() {
```

```
    }

    int ping(int t) {
        q.push(t);
        while(t-q.front()>3000)
        {
            q.pop();
        }
        return q.size();
    }
};

/**
 * Your RecentCounter object will be instantiated and called as such:
 * RecentCounter* obj = new RecentCounter();
 * int param_1 = obj->ping(t);
 */
```