

# LeetCode 933

*Number of Recent Calls*

---

Prof. Edson Alves – UnB/FGA

# Problema

Write a class `RecentCounter` to count recent requests.

It has only one method: `ping(int t)`, where  $t$  represents some time in milliseconds.

Return the number of pings that have been made from 3000 milliseconds ago until now.

Any ping with time in  $[t - 3000, t]$  will count, including the current ping.

It is guaranteed that every call to `ping` uses a strictly larger value of  $t$  than before.

## Example 1:

**Input:** `inputs = ["RecentCounter","ping","ping","ping","ping"], inputs =  
[[],[1],[100],[3001],[3002]]`

**Output:** `[null,1,2,3,3]`

## Note:

1. Each test case will have at most 10000 calls to ping.
2. Each test case will call ping with strictly increasing values of  $t$ .
3. Each call to ping will have  $1 \leq t \leq 10^9$ .

## Solução com complexidade $O(N)$

- Seja  $N$  o número de chamadas do método `ping()`
- Para determinar quantas chamadas se encontram no intervalo  $[t, t + 3000]$ , é preciso manter uma fila com os valores de  $t$  já invocados
- O primeiro elemento da fila conterá o valor  $t_a$  da chamada mais antiga
- O último elemento da fila será o valor  $t_b$  da chamada atual
- Enquanto  $t_b - t_a > 3000$ , o primeiro elemento da fila deve ser descartado
- O retorno da chamada deve ser o tamanho da fila
- Como cada elemento será inserido ou removido uma única vez, a complexidade desta solução é  $O(N)$

## Solução AC com complexidade $O(N)$

```
1 class RecentCounter {
2 public:
3     queue<int> q;
4
5     RecentCounter() {
6
7     }
8
9     int ping(int t) {
10
11         q.push(t);
12
13         while (not q.empty() and t - q.front() > 3000)
14             q.pop();
15
16         return q.size();
17     }
18 };
```