Problem 933: Number of Recent Calls

Problem Information

Difficulty: Easy
Acceptance Rate: 77.88%
Paid Only: No
Tags: Design, Queue, Data Stream

Problem Description

You have a `RecentCounter` class which counts the number of recent requests within a certain time frame.

Implement the `RecentCounter` class:

* `RecentCounter()` Initializes the counter with zero recent requests. * `int ping(int t)` Adds a new request at time `t`, where `t` represents some time in milliseconds, and returns the number of requests that has happened in the past `3000` milliseconds (including the new request). Specifically, return the number of requests that have happened in the inclusive range `[t - 3000, t]`.

It is **guaranteed** that every call to `ping` uses a strictly larger value of `t` than the previous call.

Example 1:

Input ["RecentCounter", "ping", "ping", "ping", "ping"] [[], [1], [100], [3001], [3002]] **Output** [null, 1, 2, 3, 3] **Explanation** RecentCounter recentCounter = new RecentCounter(); recentCounter.ping(1); // requests = [_1_], range is [-2999,1], return 1 recentCounter.ping(100); // requests = [_1_ , _100_], range is [-2900,100], return 2 recentCounter.ping(3001); // requests = [_1_ , _100_ , _3001_], range is [1,3001], return 3 recentCounter.ping(3002); // requests = [1, _100_ , _3001_ , _3002_], range is [2,3002], return 3

Constraints:

* `1 <= t <= 109` * Each test case will call `ping` with **strictly increasing** values of `t`. * At most `104` calls will be made to `ping`.

Code Snippets

C++:

```
class RecentCounter {
public:
RecentCounter() {

}

int ping(int t) {

}
};

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

Java:

```
class RecentCounter {

public RecentCounter() {

}

public int ping(int t) {

}
}

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

```
*/
```

Python3:

```
class RecentCounter:

def __init__(self):


def ping(self, t: int) -> int:



# Your RecentCounter object will be instantiated and called as such:
# obj = RecentCounter()
# param_1 = obj.ping(t)
```