Logger Rate Limiter

Try to solve the Logger Rate Limiter problem.

We'll cover the following

Statement

For the given stream of message requests and their timestamps as input, you must implement a logger rate limiter system that decides whether the current message request is displayed. The decision depends on whether the same message has already been displayed in the last S seconds. If yes, then the decision is FALSE, as this message is considered a duplicate. Otherwise, the decision is TRUE.

> **Note:** Several message requests, though received at different timestamps, may carry identical messages.

Constraint:

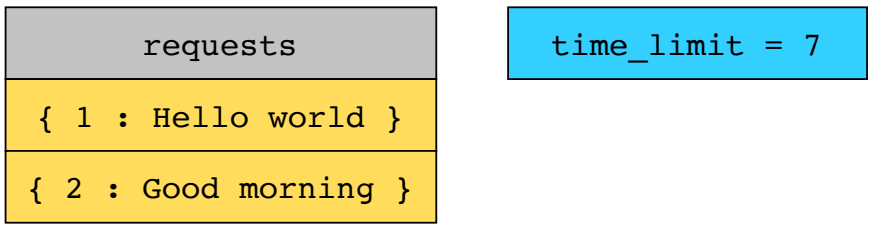
- Timestamps are in ascending order.

Examples

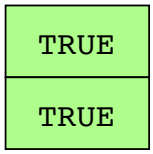
> **Note:** In the following examples, the time limit, S, is set to 7.

Sample example 3

Input



Output



As the **time_limit** is 7, the first incoming request will return TRUE. The message in the second incoming request is different from the message in the first request, hence it will also be accepted and TRUE is returned.

3 of 3

Understand the problem

Let's take a moment to make sure you've correctly understood the problem. The quiz below helps you check if you're solving the correct problem:

Logger Rate Limiter

1 Suppose the time limit for the following timestamps is 5. What will be the correct output sequence?

time stamps	message request
1	hello
4	bye
5	bye
10	hello
11	bye
14	hello

A)
TRUE
FALSE
FALSE
TRUE
FALSE
FALSE

B)
TRUE
TRUE
FALSE
TRUE
TRUE
FALSE

C)
TRUE
FALSE
FALSE
TRUE
FALSE
FALSE

Submit Answer

Question 1 of 2
0 attempted

Reset Quiz

Figure it out!

We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.

Drag and drop the cards to rearrange them in the correct sequence.

Use all incoming messages as keys and their respective timestamps as values, to form key-value pairs to store them in a hash map.

When a request arrives, use the hash map to check if it's a new request or a repeated request. If it's a new request, accept it and add it to the hash map.

If it's a repeated request, check if S seconds have passed since the last request with the same message. If this is the case, accept it and update the timestamp for that specific message in the hash map.

If the repeated request has arrived before the time limit has expired, reject it.

Reset | Show Solution | Submit

Try it yourself

Implement your solution in the following coding playground:

Java

usercode > RequestLogger.java

```java
import java.util.*;

class RequestLogger {
    // initailization of requests hash map
    public RequestLogger(int timeLimit) {
        // Write your code here
    }

    // function to accept and deny message requests
    public boolean messageRequestDecision(int timestamp, String request) {
        // checking whether the specific request exists in
        // the hash map or not

        // Your code will replace this placeholder return statement

        return false;
    }
}
```

Powered by AI | Submit

Test Cases | Results

Case 1 | Case 2 | Case 3

Input #1

[[1,"good morning"],[5,"good morning"],[9,"i need coffee"],[10,"hello world"],[11,"good mor

Logger Rate Limiter

Back

Solution: Fraction to R...

Next

Solution: Logger Rate ...

Mark as Completed