**Justin\_Design\_HashTable\_0359.**  **Logger Rate Limiter**

**Concept:**

看是否存在 hash-table 裡面

不存在則直接加入

存在則查看時間是否間格 10 秒以上了

**Code:**

class Logger:

def \_\_init\_\_(self):

"""

Initialize your data structure here.

"""

self.list\_message = []

self.list\_timestamp = []

def shouldPrintMessage(self, timestamp: int, message: str) -> bool:

"""

Returns true if the message should be printed in the given timestamp, otherwise returns false.

If this method returns false, the message will not be printed.

The timestamp is in seconds granularity.

"""

if message not in self.list\_message:

self.list\_message.append(message)

self.list\_timestamp.append(timestamp)

return True

else:

if timestamp - self.list\_timestamp[self.list\_message.index(message)] >= 10:

self.list\_timestamp[self.list\_message.index(message)] = timestamp

return True

else:

return False

# Your Logger object will be instantiated and called as such:

# obj = Logger()

# param\_1 = obj.shouldPrintMessage(timestamp,message)# param\_1 = obj.next(val)