

Design

DescriptionAcceptedEditorialSubmissionsSolutions

379. Design Phone DirectoryPremium

Solved

MediumTopicsCompanies

Design a phone directory that initially has `maxNumbers` empty slots that can store numbers. The directory should store numbers, check if a certain slot is empty or not, and empty a given slot.

Implement the `PhoneDirectory` class:

• `PhoneDirectory(int maxNumbers)` Initializes the phone directory with the number of available slots `maxNumbers`.

• `int get()` Provides a number that is not assigned to anyone. Returns `-1` if no number is available.

• `bool check(int number)` Returns `true` if the slot `number` is available and `false` otherwise.

• `void release(int number)` Recycles or releases the slot `number`.

Example 1:

Input

["PhoneDirectory", "get", "get", "check", "get", "check", "release", "check"]
[[3], [], [], [2], [], [2], [2], [2]]

Output

[null, 0, 1, true, 2, false, null, true]

Explanation

PhoneDirectory phoneDirectory = new PhoneDirectory(3);
phoneDirectory.get(); // It can return any available phone number. Here we assume it returns 0.
phoneDirectory.get(); // Assume it returns 1.
phoneDirectory.check(2); // The number 2 is available, so return true.
phoneDirectory.get(); // It returns 2, the only number that is left.
phoneDirectory.check(2); // The number 2 is no longer available, so return false.
phoneDirectory.release(2); // Release number 2 back to the pool.
phoneDirectory.check(2); // Number 2 is available again, return true.

Constraints:

• $1 \leq \text{maxNumbers} \leq 10^4$

• $0 \leq \text{number} < \text{maxNumbers}$

• At most 2×10^4 calls will be made to `get`, `check`, and `release`.

Seen this question in a real interview before? 1/5

Yes

No

Accepted 65.9K

Submissions 127.4K

Acceptance Rate 51.7%

Topics

Companies

Similar Questions

Discussion (10)

Copyright © 2024 LeetCode All rights reserved

350

10

☆

</> Code

Python3Auto

1class PhoneDirectory:

2

3def __init__(self, maxNumbers: int):

4self.__slots = set(range(maxNumbers))

5print(self.__slots)

6

7

8def get(self) -> int:

9if not self.__slots:

10return -1

11return self.__slots.pop()

12

13

14def check(self, number: int) -> bool:

15return number in self.__slots

16

17def release(self, number: int) -> None:

18self.__slots.add(number)

19

20

21

Saved

Ln 7, Col 1

TestcaseTest Result

AcceptedRuntime: 42 ms

Case 1

Input

["PhoneDirectory", "get", "get", "check", "get", "check", "release", "check"]

[[3], [], [], [2], [], [2], [2], [2]]

Stdout

{0, 1, 2}

Output