

Problem List

DescriptionEditorialSolutionsSubmissions

170. Two Sum III - Data structure design

Premium

Solved

EasyTopicsCompanies

Design a data structure that accepts a stream of integers and checks if it has a pair of integers that sum up to a particular value.

Implement the `TwoSum` class:

TwoSum()

Initializes the `TwoSum` object with an empty array initially.

void add(int number)

Adds `number` to the data structure.

boolean find(int value)

Returns `true` if there exists any pair of numbers whose sum is equal to `value`, otherwise, it returns `false`.

Example 1:

Input

["TwoSum", "add", "add", "add", "find", "find"]
[[], [1], [3], [5], [4], [7]]

Output

[null, null, null, null, true, false]

Explanation

TwoSum twoSum = new TwoSum();
twoSum.add(1); // [] --> [1]
twoSum.add(3); // [1] --> [1,3]
twoSum.add(5); // [1,3] --> [1,3,5]
twoSum.find(4); // 1 + 3 = 4, return true
twoSum.find(7); // No two integers sum up to 7, return false

Constraints:

-10⁵ <= number <= 10⁵

-2³¹ <= value <= 2³¹ - 1

At most 10⁴ calls will be made to `add` and `find`.

Seen this question in a real interview before?

1/5

YesNo

Accepted 153.9K | Submissions 404.8K | Acceptance Rate 38.0%

Topics

Companies

Similar Questions

Discussion (7)

Copyright © 2024 LeetCode All rights reserved

6667

</> Code

Python3Auto

1class TwoSum:

2

3def __init__(self):

4self.ans = []

5

6

7def add(self, number: int) -> None:

8self.ans.append(number)

9

10

11def find(self, value: int) -> bool:

12

13seen = {}

14for i in range(len(self.ans)):

15diff = value - self.ans[i]

16if diff in seen:

17return True

18else:

19seen[self.ans[i]] = i

20return False

21

22

23

24# Your TwoSum object will be instantiated and called as such:

25# obj = TwoSum()

26# obj.add(number)

27# param_2 = obj.find(value)

SavedLn 10, Col 1

TestcaseTest Result

AcceptedRuntime: 37 ms

Case 1

Input

["TwoSum", "add", "add", "add", "find", "find"]

[[], [1], [3], [5], [4], [7]]