

0170. 两数之和 III - 数据结构设计

👤 ITCharge ⌚ 大约 1 分钟

- 标签：设计、数组、哈希表、双指针、数据流
- 难度：简单

题目链接

- [0170. 两数之和 III - 数据结构设计 - 力扣](#)

题目大意

设计一个接受整数流的数据结构，使该数据结构支持检查是否存在两数之和等于特定值。

实现 TwoSum 类：

- `TwoSum()`：使用空数组初始化 TwoSum 对象
- `def add(self, number: int) -> None` 向数据结构添加一个数 `number`
- `def find(self, value: int) -> bool` 寻找数据结构中是否存在一对整数，使得两数之和与给定的值 `value` 相等。如果存在，返回 `True`；否则，返回 `False`。

解题思路

使用哈希表存储数组元素值与元素频数的关系。哈希表中键值对信息为 `number: count`。
`count` 为 `number` 在数组中的频数。

- `add(number)` 函数中：在哈希表添加 `number` 与其频数之间的关系。
- `find(number)` 函数中：遍历哈希表，对于每个 `number`，检测哈希表中是否存在 `value - number`，如果存在则终止循环并返回结果。
 - 如果 `number == value - number`，则判断哈希表中 `number` 的数目是否大于等于 2。

代码

py

```
class TwoSum:

    def __init__(self):
        """
        Initialize your data structure here.
        """
        self.num_counts = dict()

    def add(self, number: int) -> None:
        """
        Add the number to an internal data structure..
        """
        if number in self.num_counts:
            self.num_counts[number] += 1
        else:
            self.num_counts[number] = 1

    def find(self, value: int) -> bool:
        """
        Find if there exists any pair of numbers which sum is equal to the
        value.
        """
        for number in self.num_counts.keys():
            number2 = value - number
            if number == number2:
                if self.num_counts[number] > 1:
                    return True
            else:
                if number2 in self.num_counts:
                    return True
        return False
```

