Problem 170: Two Sum III - Data structure design

Problem Information

Difficulty: Easy
Acceptance Rate: 38.98%
Paid Only: Yes
Tags: Array, Hash Table, Two Pointers, Design, Data Stream

Problem Description

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`.

Code Snippets

C++:

```
class TwoSum {
public:
TwoSum() {

}

void add(int number) {

}

bool find(int value) {

}
};

/**
* Your TwoSum object will be instantiated and called as such:
* TwoSum* obj = new TwoSum();
* obj->add(number);
* bool param_2 = obj->find(value);
*/
```

Java:

```
class TwoSum {

public TwoSum() {

}

public void add(int number) {

}

public boolean find(int value) {

}
}

/**
* Your TwoSum object will be instantiated and called as such:
```

```
* TwoSum obj = new TwoSum();
* obj.add(number);
* boolean param_2 = obj.find(value);
*/
```

Python3:

```
class TwoSum:

def __init__(self):



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



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




# Your TwoSum object will be instantiated and called as such:
# obj = TwoSum()
# obj.add(number)
# param_2 = obj.find(value)
```