# Problem 1429: First Unique Number

## Problem Information

**Difficulty:** Medium
**Acceptance Rate:** 56.60%
**Paid Only:** Yes
**Tags:** Array, Hash Table, Design, Queue, Data Stream

## Problem Description

You have a queue of integers, you need to retrieve the first unique integer in the queue.

Implement the `FirstUnique` class:

* `FirstUnique(int[] nums)` Initializes the object with the numbers in the queue. * `int showFirstUnique()` returns the value of **the first unique** integer of the queue, and returns **-1** if there is no such integer. * `void add(int value)` insert value to the queue.

**Example 1:**

**Input:** ["FirstUnique","showFirstUnique","add","showFirstUnique","add","showFirstUnique","add","showFirstUnique"] [[[2,3,5]],[],[5],[],[2],[],[3],[]] **Output:** [null,2,null,2,null,3,null,-1] **Explanation:** FirstUnique firstUnique = new FirstUnique([2,3,5]); firstUnique.showFirstUnique(); // return 2 firstUnique.add(5); // the queue is now [2,3,5,5] firstUnique.showFirstUnique(); // return 2 firstUnique.add(2); // the queue is now [2,3,5,5,2] firstUnique.showFirstUnique(); // return 3 firstUnique.add(3); // the queue is now [2,3,5,5,2,3] firstUnique.showFirstUnique(); // return -1

**Example 2:**

**Input:** ["FirstUnique","showFirstUnique","add","add","add","add","add","showFirstUnique"] [[[7,7,7,7,7,7]],[],[7],[3],[3],[7],[17],[]] **Output:** [null,-1,null,null,null,null,null,17] **Explanation:** FirstUnique firstUnique = new FirstUnique([7,7,7,7,7,7]); firstUnique.showFirstUnique(); // return -1 firstUnique.add(7); // the queue is now [7,7,7,7,7,7,7] firstUnique.add(3); // the queue is now [7,7,7,7,7,7,7,3] firstUnique.add(3); // the queue is now [7,7,7,7,7,7,7,3,3] firstUnique.add(7); // the queue is now [7,7,7,7,7,7,7,3,3,7]

firstUnique.add(17); // the queue is now [7,7,7,7,7,7,7,3,3,7,17] firstUnique.showFirstUnique(); // return 17

**Example 3:**

**Input:** ["FirstUnique","showFirstUnique","add","showFirstUnique"] [[[809]],[],[809],[]] **Output:** [null,809,null,-1] **Explanation:** FirstUnique firstUnique = new FirstUnique([809]); firstUnique.showFirstUnique(); // return 809 firstUnique.add(809); // the queue is now [809,809] firstUnique.showFirstUnique(); // return -1

**Constraints:**

* `1 <= nums.length <= 10^5` * `1 <= nums[i] <= 10^8` * `1 <= value <= 10^8` * At most `50000` calls will be made to `showFirstUnique` and `add`.

## Code Snippets

**C++:**

```
class FirstUnique {
public:
FirstUnique(vector<int>& nums) {

}

int showFirstUnique() {

}

void add(int value) {

}
};

/**
* Your FirstUnique object will be instantiated and called as such:
* FirstUnique* obj = new FirstUnique(nums);
* int param_1 = obj->showFirstUnique();
* obj->add(value);
*/
```

**Java:**

```
class FirstUnique {

public FirstUnique(int[] nums) {

}

public int showFirstUnique() {

}

public void add(int value) {

}
}

/**
* Your FirstUnique object will be instantiated and called as such:
* FirstUnique obj = new FirstUnique(nums);
* int param_1 = obj.showFirstUnique();
* obj.add(value);
*/
```

**Python3:**

```
class FirstUnique:

def __init__(self, nums: List[int]):


def showFirstUnique(self) -> int:


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



# Your FirstUnique object will be instantiated and called as such:
# obj = FirstUnique(nums)
# param_1 = obj.showFirstUnique()
# obj.add(value)
```