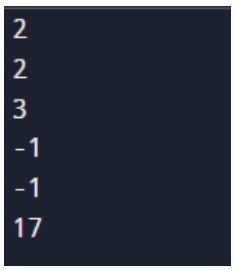
34. First Unique Number You have a queue of integers, you need to retrieve the first unique integer in the queue. Implement the FirstUnique class: ● FirstUnique(int[] nums) Initializes the object with the numbers in the queue. ● int showFirstUnique() returns the value of the first unique integer of the 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","a
dd","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

PROGRAM:

```
from collections import OrderedDict


class FirstUnique:
    def __init__(self, nums):
        self.queue = OrderedDict()  # To maintain insertion order
        self.unique_nums = {}  # To store frequency of numbers
        for num in nums:
            self.add(num)


    def showFirstUnique(self):
        if self.queue:
            return next(iter(self.queue))
        return -1


    def add(self, value):
```

```
        if value in self.unique_nums:
            if self.unique_nums[value]:
                self.queue.pop(value, None)
                self.unique_nums[value] = False
        else:
            self.queue[value] = None
            self.unique_nums[value] = True
firstUnique1 = FirstUnique([2, 3, 5])
print(firstUnique1.showFirstUnique())
firstUnique1.add(5)
print(firstUnique1.showFirstUnique())
firstUnique1.add(2)
print(firstUnique1.showFirstUnique())
firstUnique1.add(3)
print(firstUnique1.showFirstUnique())

firstUnique2 = FirstUnique([7, 7, 7, 7, 7, 7])
print(firstUnique2.showFirstUnique())
firstUnique2.add(7)
firstUnique2.add(3)
firstUnique2.add(3)
firstUnique2.add(7)
firstUnique2.add(17)
print(firstUnique2.showFirstUnique())
```

OUTPUT:



TIME COMPLEXITY: O(1)