4. 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

PROGRAM:-

```
from collections import deque, defaultdict

class FirstUnique:
    def _init_(self, nums):
        self.queue = deque(nums)  # Queue to maintain order of elements
        self.count = defaultdict(int)  # Dictionary to count occurrences
        for num in nums:
            self.count[num] += 1

    def showFirstUnique(self):
        # Remove elements from the front of the queue until we find a unique one
        while self.queue and self.count[self.queue[0]] > 1:
            self.queue.popleft()
        return self.queue[0] if self.queue else -1

    def add(self, value):
        self.queue.append(value)
        self.count[value] += 1

# Example usage
firstUnique = FirstUnique([2, 3, 5])
print(firstUnique.showFirstUnique())  # return 2
firstUnique.add(5)  # the queue is now [2, 3, 5, 5]
print(firstUnique.showFirstUnique())  # return 2
firstUnique.add(2)  # the queue is now [2, 3, 5, 5, 2]
print(firstUnique.showFirstUnique())  # return 3
firstUnique.add(3)  # the queue is now [2, 3, 5, 5, 2, 3]
print(firstUnique.showFirstUnique())  # return -1
```

OUTPUT:--

TIME COMPLEXITY:-O(n)