Dashboard / My courses / CS23331-DAA-2023-CSE / Divide and Conquer / 2-Majority Element

Started on	Friday, 20 September 2024, 2:47 PM
State	Finished
Completed on	Friday, 20 September 2024, 2:48 PM
Time taken	39 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

Question **1**

Correct

Mark 1.00 out of 1.00

Given an array nums of size n, return *the majority element*.

The majority element is the element that appears more than ⌊n / 2⌋ times. You may assume that the majority element always exists in the array.

Example 1:

```
Input: nums = [3,2,3]
Output: 3
```

Example 2:

```
Input: nums = [2,2,1,1,1,2,2]
Output: 2
```

Constraints:

- n == nums.length
- 1 <= n <= 5 * 10^4
- -2^{31} <= nums[i] <= 2^{31} - 1

For example:

Input	Result
3 3 2 3	3
7 2 2 1 1 1 2 2	2

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
#include <stdlib.h>
int Majority(int nums[], int size)
{
    int count = 0;
    int candidate = nums[0];
    for (int i = 0; i < size; i++)
    {
        if (count == 0)
        {
            candidate = nums[i];
        }
        if (nums[i] == candidate)
        {
            count++;
        }
        else
        {
            count--;
        }
    }
    count = 0;
    for (int i = 0; i < size; i++)
    {
        if (nums[i] == candidate)
        {
            count++;
        }
    }
    if (count > size / 2)
    {
        return candidate;
    }
    return -1;
}
int main()
{
    int n;
    scanf("%d", &n);
    int *nums = (int *)malloc(n * sizeof(int));
    if (nums == NULL)
    {
        return 1;
    }
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &nums[i]);
    }
    int mElement = Majority(nums, n);
    printf("%d\n", mElement);
    free(nums);
    return 0;
```

	Input	Expected	Got	
✓	3 3 2 3	3	3	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

◄ 1-Number of Zeros in a Given Array

Jump to...

3-Finding Floor Value ►