

[Array](#) [Matrix](#) [Strings](#) [Hashing](#) [Linked List](#) [Stack](#) [Queue](#) [Binary Tree](#) [Binary Search](#)

# Majority Element

Difficulty Level : Medium • Last Updated : 06 Jul, 2022



Write a function which takes an array and prints the majority element (if it exists), otherwise prints "No Majority Element". A **majority element** in an array  $A[]$  of size  $n$  is an element that appears more than  $n/2$  times (and hence there is at most one such element).

## Examples :

Input : {3, 3, 4, 2, 4, 4, 2, 4, 4}



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

[GET \\$100 FREE CREDIT](#)

[HIDE AD](#) • [AD VIA BUYSPELLADS](#)

than the half of the size of the array size.

**Input :** {3, 3, 4, 2, 4, 4, 2, 4}

**Output :** No Majority Element

**Explanation:** There is no element whose frequency is greater than the half of the size of the array size.

Recommended Practice

**Majority Element**

Try It!

## METHOD 1

- **Approach:** The basic solution is to have two loops and keep track of the maximum count for all different elements. If maximum count becomes greater than  $n/2$  then break the loops and return the element having maximum count. If the maximum count doesn't become more than  $n/2$  then the majority element doesn't exist.
- **Algorithm:**
  1. Create a variable to store the max count, *count* = 0
  2. Traverse through the array from start to end.
  3. For every element in the array run another loop to find the count of similar elements in the given array.
  4. If the count is greater than the max count update the max count and store the index in another variable.
  5. If the maximum count is greater than the half the size of the array, print the element. Else print there is no majority element.



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS



programming knowledge in just 6 months and ace coding interviews at top tech companies.

[Explore Now](#)

Below is the implementation of the above idea:

## C++

```
// C++ program to find Majority
// element in an array
#include <bits/stdc++.h>
using namespace std;

// Function to find Majority element
// in an array
void findMajority(int arr[], int n)
{
    int maxCount = 0;
    int index = -1; // sentinels
    for (int i = 0; i < n; i++) {
        int count = 0;
        for (int j = 0; j < n; j++) {
            if (arr[i] == arr[j])
                count++;
        }

        // update maxCount if count of
        // current element is greater
        if (count > maxCount) {
            maxCount = count;
            index = i;
        }
    }

    // if maxCount is greater than n/2
    // return the corresponding element
    if (maxCount > n / 2)
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

[GET \\$100 FREE CREDIT](#)

HIDE AD • AD VIA BUYSPELLADS

```
}

// Driver code
int main()
{
    int arr[] = { 1, 1, 2, 1, 3, 5, 1 };
    int n = sizeof(arr) / sizeof(arr[0]);

    // Function calling
    findMajority(arr, n);

    return 0;
}
```

## Java

```
// Java program to find Majority
// element in an array

import java.io.*;

class GFG {

    // Function to find Majority element
    // in an array
    static void findMajority(int arr[], int n)
    {
        int maxCount = 0;
        int index = -1; // sentinels
        for (int i = 0; i < n; i++) {
            int count = 0;
            for (int j = 0; j < n; j++) {
                if (arr[i] == arr[j])
                    count++;
            }

            // update maxCount if count of
            // current element is greater
            if (count > maxCount) {
                maxCount = count;
            }
        }
    }
}
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
// if maxCount is greater than n/2
// return the corresponding element
if (maxCount > n / 2)
    System.out.println(arr[index]);

else
    System.out.println("No Majority Element");
}

// Driver code
public static void main(String[] args)
{

    int arr[] = { 1, 1, 2, 1, 3, 5, 1 };
    int n = arr.length;

    // Function calling
    findMajority(arr, n);
}
// This code is contributed by ajit.
}
```

## Python3

```
# Python3 program to find Majority
# element in an array
```

```
# Function to find Majority
# element in an array
```

```
def findMajority(arr, n):
```

```
    maxCount = 0
    index = -1 # sentinels
    for i in range(n):
```

```
        count = 0
        for j in range(n):
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
# update maxCount if count of
# current element is greater
if(count > maxCount):

    maxCount = count
    index = i

# if maxCount is greater than n/2
# return the corresponding element
if (maxCount > n//2):
    print(arr[index])

else:
    print("No Majority Element")

# Driver code
if __name__ == "__main__":
    arr = [1, 1, 2, 1, 3, 5, 1]
    n = len(arr)

    # Function calling
    findMajority(arr, n)

# This code is contributed
# by ChitraNayal
```

## C#

```
// C# program to find Majority
// element in an array
using System;

public class GFG {

    // Function to find Majority element
    // in an array
    static void findMajority(int[] arr, int n)
    {
        int maxCount = 0;
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
        for (int j = 0; j < n; j++) {
            if (arr[i] == arr[j])
                count++;
        }

        // update maxCount if count of
        // current element is greater
        if (count > maxCount) {
            maxCount = count;
            index = i;
        }
    }

    // if maxCount is greater than n/2
    // return the corresponding element
    if (maxCount > n / 2)
        Console.WriteLine(arr[index]);

    else
        Console.WriteLine("No Majority Element");
}

// Driver code
static public void Main()
{
    int[] arr = { 1, 1, 2, 1, 3, 5, 1 };
    int n = arr.Length;

    // Function calling
    findMajority(arr, n);
}
// This code is contributed by Tushil..
}
```

## PHP

```
<?php
// PHP program to find Majority
// element in an array
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
{
    $maxCount = 0;
    $index = -1; // sentinels
    for($i = 0; $i < $n; $i++)
    {
        $count = 0;
        for($j = 0; $j < $n; $j++)
        {
            if($arr[$i] == $arr[$j])
                $count++;
        }

        // update maxCount if count of
        // current element is greater
        if($count > $maxCount)
        {
            $maxCount = $count;
            $index = $i;
        }
    }

    // if maxCount is greater than n/2
    // return the corresponding element
    if ($maxCount > $n/2)
        echo $arr[$index] . "\n";
    else
        echo "No Majority Element" . "\n";
}

// Driver code
$arr = array(1, 1, 2, 1, 3, 5, 1);
$n = sizeof($arr);

// Function calling
findMajority($arr, $n);

// This code is contributed
// by Akanksha Rai
```

## Javascript



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS



```
// Javascript program to find Majority
// element in an array

// Function to find Majority element
// in an array
function findMajority(arr, n)
{
    let maxCount = 0;
    let index = -1; // sentinels

    for(let i = 0; i < n; i++)
    {
        let count = 0;
        for(let j = 0; j < n; j++)
        {
            if (arr[i] == arr[j])
                count++;

            // Update maxCount if count of
            // current element is greater
            if (count > maxCount)
            {
                maxCount = count;
                index = i;
            }
        }
    }

    // If maxCount is greater than n/2
    // return the corresponding element
    if (maxCount > n / 2)
        document.write(arr[index]);
    else
        document.write("No Majority Element");
}

// Driver code
let arr = [ 1, 1, 2, 1, 3, 5, 1 ];
let n = arr.length;

// Function calling
findMajority(arr, n);
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
    /script>
```

## Output

1

## Complexity Analysis:

- **Time Complexity:**  $O(n^2)$ .

A nested loop is needed where both the loops traverse the array from start to end, so the time complexity is  $O(n^2)$ .

- **Auxiliary Space:**  $O(1)$ .

As no extra space is required for any operation so the space complexity is constant.

## METHOD 2 (Using Binary Search Tree)

- **Approach:** Insert elements in BST one by one and if an element is already present then increment the count of the node. At any stage, if the count of a node becomes more than  $n/2$  then return.

- **Algorithm:**

1. Create a binary search tree, if same element is entered in the binary search tree the frequency of the node is increased.
2. traverse the array and insert the element in the binary search tree.
3. If the maximum frequency of any node is greater than the half the size of the array, then perform a inorder traversal and find the node with frequency greater than half
4. Else print No majority Element.

Below is the implementation of the above idea:



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
// search tree.
#include <bits/stdc++.h>
using namespace std;

struct node {
    int key;
    int c = 0;
    struct node *left, *right;
};

// A utility function to create a new BST node
struct node* newNode(int item)
{
    struct node* temp
        = (struct node*)malloc(sizeof(struct node));
    temp->key = item;
    temp->c = 1;
    temp->left = temp->right = NULL;
    return temp;
}

// A utility function to insert a new node with given key in
// BST
struct node* insert(struct node* node, int key, int& ma)
{
    // If the tree is empty, return a new node
    if (node == NULL) {
        if (ma == 0)
            ma = 1;

        return newNode(key);
    }

    // Otherwise, recur down the tree
    if (key < node->key)
        node->left = insert(node->left, key, ma);
    else if (key > node->key)
        node->right = insert(node->right, key, ma);
    else
        node->c++;

    // find the max count
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```

    return node;
}

// A utility function to do inorder traversal of BST
void inorder(struct node* root, int s)
{
    if (root != NULL) {
        inorder(root->left, s);

        if (root->c > (s / 2))
            printf("%d \n", root->key);

        inorder(root->right, s);
    }
}

// Driver Code
int main()
{
    int a[] = { 1, 3, 3, 3, 2 };
    int size = (sizeof(a)) / sizeof(a[0]);

    struct node* root = NULL;
    int ma = 0;

    for (int i = 0; i < size; i++) {
        root = insert(root, a[i], ma);
    }

    // Function call
    if (ma > (size / 2))
        inorder(root, size);
    else
        cout << "No majority element\n";
    return 0;
}

```

## Java

```

// Java program to demonstrate insert
// operation in binary search tree.
import java.io.*;

```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
    int key;
    int c = 0;
    Node left, right;

}

class GFG{

static int ma = 0;

// A utility function to create a
// new BST node
static Node newNode(int item)
{
    Node temp = new Node();
    temp.key = item;
    temp.c = 1;
    temp.left = temp.right = null;
    return temp;
}

// A utility function to insert a new node
// with given key in BST
static Node insert(Node node, int key)
{
    // If the tree is empty,
    // return a new node
    if (node == null)
    {
        if (ma == 0)
            ma = 1;

        return newNode(key);
    }

    // Otherwise, recur down the tree
    if (key < node.key)
        node.left = insert(node.left, key);
    else if (key > node.key)
        node.right = insert(node.right, key);
    else
        node.c++;
}
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
// Return the (unchanged) node pointer
return node;
}

// A utility function to do inorder
// traversal of BST
static void inorder(Node root, int s)
{
    if (root != null)
    {
        inorder(root.left, s);

        if (root.c > (s / 2))
            System.out.println(root.key + "\n");

        inorder(root.right, s);
    }
}

// Driver Code
public static void main(String[] args)
{
    int a[] = { 1, 3, 3, 3, 2 };
    int size = a.length;
    Node root = null;

    for(int i = 0; i < size; i++)
    {
        root = insert(root, a[i]);
    }

    // Function call
    if (ma > (size / 2))
        inorder(root, size);
    else
        System.out.println("No majority element\n");
}
}

// This code is contributed by avanitrachhadiya2155
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
# Python3 program to demonstrate insert operation in binary
# search tree.
# class for creating node
class Node():
    def __init__(self, data):
        self.data = data
        self.left = None
        self.right = None
        self.count = 1 # count of number of times data is inserted in tree

# class for binary search tree
# it initialises tree with None root
# insert function inserts node as per BST rule
# and also checks for majority element
# if no majority element is found yet, it returns None

class BST():
    def __init__(self):
        self.root = None

    def insert(self, data, n):
        out = None
        if (self.root == None):
            self.root = Node(data)
        else:
            out = self.insertNode(self.root, data, n)
        return out

    def insertNode(self, currentNode, data, n):
        if (currentNode.data == data):
            currentNode.count += 1
            if (currentNode.count > n//2):
                return currentNode.data
            else:
                return None
        elif (currentNode.data < data):
            if (currentNode.right):
                self.insertNode(currentNode.right, data, n)
            else:
                currentNode.right = Node(data)
        elif (currentNode.data > data):
```



A quick, easy, and intuitive way to build, deploy,  
manage, and scale your apps.

**GET \$100 FREE CREDIT**

HIDE AD • AD VIA BUYSPELLADS

```
currentNode.left = Node(data)
```

```
# Driver code
# declaring an array
arr = [3, 2, 3]
n = len(arr)

# declaring None tree
tree = BST()
flag = 0
for i in range(n):
    out = tree.insert(arr[i], n)
    if (out != None):
        print(arr[i])
        flag = 1
        break
if (flag == 0):
    print("No Majority Element")
```

## C#

```
// C# program to demonstrate insert
// operation in binary search tree.
using System;

public class Node
{
    public int key;
    public int c = 0;
    public Node left, right;
}

class GFG{

    static int ma = 0;

    // A utility function to create a
    // new BST node
    static Node newNode(int item)
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS



```
temp.c = 1;
temp.left = temp.right = null;
return temp;
}

// A utility function to insert a new node
// with given key in BST
static Node insert(Node node, int key)
{
    // If the tree is empty,
    // return a new node
    if (node == null)
    {
        if (ma == 0)
            ma = 1;

        return newNode(key);
    }

    // Otherwise, recur down the tree
    if (key < node.key)
        node.left = insert(node.left, key);
    else if (key > node.key)
        node.right = insert(node.right, key);
    else
        node.c++;

    // Find the max count
    ma = Math.Max(ma, node.c);

    // Return the (unchanged) node pointer
    return node;
}

// A utility function to do inorder
// traversal of BST
static void inorder(Node root, int s)
{
    if (root != null)
    {
        inorder(root.left, s);
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```

        inorder(root.right, s);
    }
}

// Driver Code
static public void Main()
{
    int[] a = { 1, 3, 3, 3, 2 };
    int size = a.Length;
    Node root = null;

    for(int i = 0; i < size; i++)
    {
        root = insert(root, a[i]);
    }

    // Function call
    if (ma > (size / 2))
        inorder(root, size);
    else
        Console.WriteLine("No majority element\n");
}
}

// This code is contributed by rag2127

```

## Javascript

```

<script>
// javascript program to demonstrate insert
// operation in binary search tree.
class Node {
    constructor(){
        this.key = 0;
        this.c = 0;
        this.left = null,
        this.right = null;
    }
}

```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
// new BST node
function newNode(item)
{
    var temp = new Node();
    temp.key = item;
    temp.c = 1;
    temp.left = temp.right = null;
    return temp;
}

// A utility function to insert a new node
// with given key in BST
function insert(node , key) {

    // If the tree is empty,
    // return a new node
    if (node == null) {
        if (ma == 0)
            ma = 1;

        return newNode(key);
    }

    // Otherwise, recur down the tree
    if (key < node.key)
        node.left = insert(node.left, key);
    else if (key > node.key)
        node.right = insert(node.right, key);
    else
        node.c++;

    // Find the max count
    ma = Math.max(ma, node.c);

    // Return the (unchanged) node pointer
    return node;
}

// A utility function to do inorder
// traversal of BST
function inorder(root , s) {
    if (root != null) {
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
        document.write(root.key + "\n");

        inorder(root.right, s);
    }
}

// Driver Code

var a = [ 1, 3, 3, 3, 2 ];
var size = a.length;
var root = null;

for (i = 0; i < size; i++) {
    root = insert(root, a[i]);
}

// Function call
if (ma > (size / 2))
    inorder(root, size);
else
    document.write("No majority element\n");

// This code is contributed by gauravrajput1
</script>
```

## Output

3

## Complexity Analysis:

- **Time Complexity:** If a [Binary Search Tree](#) is used then time complexity will be  $O(n^2)$ . If a [self-balancing-binary-search](#) tree is used then it will be  $O(n \log n)$
- **Auxiliary Space:**  $O(n)$ .

As extra space is needed to store the array in tree.

## METHOD 3 (Using Moore's Voting Algorithm):



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

in the array. If there is a majority element in an array, then this step will definitely return majority element, otherwise, it will return candidate for majority element.

- Check if the element obtained from the above step is majority element. This step is necessary as there might be no majority element.

- **Algorithm:**

1. Loop through each element and maintains a count of majority element, and a majority index, *maj\_index*
2. If the next element is same then increment the count if the next element is not same then decrement the count.
3. if the count reaches 0 then changes the *maj\_index* to the current element and set the count again to 1.
4. Now again traverse through the array and find the count of majority element found.
5. If the count is greater than half the size of the array, print the element
6. Else print that there is no majority element

Below is the implementation of above idea:

---

## C++

```
// C++ Program for finding out
// majority element in an array
#include <bits/stdc++.h>
using namespace std;

/* Function to find the candidate for Majority */
int findCandidate(int a[], int size)
{
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
        count++;
    else
        count--;
    if (count == 0) {
        maj_index = i;
        count = 1;
    }
}
return a[maj_index];
}

/* Function to check if the candidate
occurs more than n/2 times */
bool isMajority(int a[], int size, int cand)
{
    int count = 0;
    for (int i = 0; i < size; i++)

        if (a[i] == cand)
            count++;

    if (count > size / 2)
        return 1;

    else
        return 0;
}

/* Function to print Majority Element */
void printMajority(int a[], int size)
{
    /* Find the candidate for Majority*/
    int cand = findCandidate(a, size);

    /* Print the candidate if it is Majority*/
    if (isMajority(a, size, cand))
        cout << " " << cand << " ";

    else
        cout << "No Majority Element";
}
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
int a[] = { 1, 3, 3, 1, 2 };
int size = (sizeof(a)) / sizeof(a[0]);

// Function calling
printMajority(a, size);

return 0;
}
```

## C

```
/* Program for finding out majority element in an array */
#include <stdio.h>
#define bool int

int findCandidate(int*, int);
bool isMajority(int*, int, int);

/* Function to print Majority Element */
void printMajority(int a[], int size)
{
    /* Find the candidate for Majority*/
    int cand = findCandidate(a, size);

    /* Print the candidate if it is Majority*/
    if (isMajority(a, size, cand))
        printf(" %d ", cand);
    else
        printf("No Majority Element");
}

/* Function to find the candidate for Majority */
int findCandidate(int a[], int size)
{
    int maj_index = 0, count = 1;
    int i;
    for (i = 1; i < size; i++) {
        if (a[maj_index] == a[i])
            count++;
        else
            count--;
    }
}
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```

    }
}
return a[maj_index];
}

/* Function to check if the candidate occurs more than n/2
 * times */
bool isMajority(int a[], int size, int cand)
{
    int i, count = 0;
    for (i = 0; i < size; i++)
        if (a[i] == cand)
            count++;
    if (count > size / 2)
        return 1;
    else
        return 0;
}

/* Driver code */
int main()
{
    int a[] = { 1, 3, 3, 1, 2 };
    int size = (sizeof(a)) / sizeof(a[0]);

    // Function call
    printMajority(a, size);
    getchar();
    return 0;
}

```

## Java

```

/* Program for finding out majority element in an array */

class MajorityElement {
    /* Function to print Majority Element */
    void printMajority(int a[], int size)
    {
        /* Find the candidate for Majority*/
        int cand = findCandidate(a, size);
    }
}

```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS



```
        System.out.println(" " + cand + " ");
    else
        System.out.println("No Majority Element");
}

/* Function to find the candidate for Majority */
int findCandidate(int a[], int size)
{
    int maj_index = 0, count = 1;
    int i;
    for (i = 1; i < size; i++) {
        if (a[maj_index] == a[i])
            count++;
        else
            count--;
        if (count == 0) {
            maj_index = i;
            count = 1;
        }
    }
    return a[maj_index];
}

/* Function to check if the candidate occurs more
than n/2 times */
boolean isMajority(int a[], int size, int cand)
{
    int i, count = 0;
    for (i = 0; i < size; i++) {
        if (a[i] == cand)
            count++;
    }
    if (count > size / 2)
        return true;
    else
        return false;
}

/* Driver code */
public static void main(String[] args)
{
    MajorityElement majorelement
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
        // Function call
        int size = a.length;
        majorelement.printMajority(a, size);
    }
}
```

// This code has been contributed by Mayank Jaiswal

## Python3

```
# Program for finding out majority element in an array

# Function to find the candidate for Majority

def findCandidate(A):
    maj_index = 0
    count = 1
    for i in range(len(A)):
        if A[maj_index] == A[i]:
            count += 1
        else:
            count -= 1
            if count == 0:
                maj_index = i
                count = 1
    return A[maj_index]

# Function to check if the candidate occurs more than n/2 times

def isMajority(A, cand):
    count = 0
    for i in range(len(A)):
        if A[i] == cand:
            count += 1
    if count > len(A)/2:
        return True
    else:
        return False
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```

def printMajority(A):
    # Find the candidate for Majority
    cand = findCandidate(A)

    # Print the candidate if it is Majority
    if isMajority(A, cand) == True:
        print(cand)
    else:
        print("No Majority Element")

# Driver code
A = [1, 3, 3, 1, 2]

# Function call
printMajority(A)

```

## C#

```

// C# Program for finding out majority element in an array
using System;

class GFG {
    /* Function to print Majority Element */
    static void printMajority(int[] a, int size)
    {
        /* Find the candidate for Majority*/
        int cand = findCandidate(a, size);

        /* Print the candidate if it is Majority*/
        if (isMajority(a, size, cand))
            Console.Write(" " + cand + " ");
        else
            Console.Write("No Majority Element");
    }

    /* Function to find the candidate for Majority */
    static int findCandidate(int[] a, int size)
    {
        int maj_index = 0, count = 1;
        int i;

```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
        else
            count--;

        if (count == 0) {
            maj_index = i;
            count = 1;
        }
    }
    return a[maj_index];
}

// Function to check if the candidate
// occurs more than n/2 times
static bool isMajority(int[] a, int size, int cand)
{
    int i, count = 0;
    for (i = 0; i < size; i++) {
        if (a[i] == cand)
            count++;
    }
    if (count > size / 2)
        return true;
    else
        return false;
}

// Driver Code
public static void Main()
{
    int[] a = { 1, 3, 3, 1, 2 };
    int size = a.Length;

    // Function call
    printMajority(a, size);
}

// This code is contributed by Sam007
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
// PHP Program for finding out majority
// element in an array

// Function to find the candidate
// for Majority
function findCandidate($a, $size)
{
    $maj_index = 0;
    $count = 1;
    for ($i = 1; $i < $size; $i++)
    {
        if ($a[$maj_index] == $a[$i])
            $count++;
        else
            $count--;
        if ($count == 0)
        {
            $maj_index = $i;
            $count = 1;
        }
    }
    return $a[$maj_index];
}

// Function to check if the candidate
// occurs more than n/2 times
function isMajority($a, $size, $cand)
{
    $count = 0;
    for ($i = 0; $i < $size; $i++)

        if ($a[$i] == $cand)
            $count++;

    if ($count > $size / 2)
        return 1;

    else
        return 0;
}

// Function to print Majority Element
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```

    $cand = findCandidate($a, $size);

    /* Print the candidate if it is Majority*/
    if (isMajority($a, $size, $cand))
        echo " ", $cand, " ";
    else
        echo "No Majority Element";
}

// Driver Code
$a = array(1, 3, 3, 1, 2);
$size = sizeof($a);

// Function calling
printMajority($a, $size);

// This code is contributed by jit_t
?>

```

## Javascript

```

<script>
    // Javascript Program for finding out majority element in an array

    /* Function to print Majority Element */
    function printMajority(a, size)
    {
        /* Find the candidate for Majority*/
        let cand = findCandidate(a, size);

        /* Print the candidate if it is Majority*/
        if (isMajority(a, size, cand))
            document.write(" " + cand + " ");
        else
            document.write("No Majority Element");
    }

    /* Function to find the candidate for Majority */
    function findCandidate(a, size)
    {
        let maj_index = 0, count = 1;

```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
        count++;
    else
        count--;

    if (count == 0) {
        maj_index = i;
        count = 1;
    }
}
return a[maj_index];
}

// Function to check if the candidate
// occurs more than n/2 times
function isMajority(a, size, cand)
{
    let i, count = 0;
    for (i = 0; i < size; i++) {
        if (a[i] == cand)
            count++;
    }
    if (count > parseInt(size / 2, 10))
        return true;
    else
        return false;
}

let a = [ 1, 3, 3, 1, 2 ];
let size = a.length;

// Function call
printMajority(a, size);

// This code is contributed by rameshtravel07.
</script>
```

## Output

No Majority Element

## Complexity Analysis:



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

As two traversal of the array is needed, so the time complexity is

- **Auxiliary Space:**  $O(1)$ .

As no extra space is required.

#### **METHOD 4 (Using Hashmap):**

- **Approach:** This method is somewhat similar to Moore voting algorithm in terms of time complexity, but in this case, there is no need for the second step of Moore voting algorithm. But as usual, here space complexity becomes  $O(n)$ .

In Hashmap(key-value pair), at value, maintain a count for each element(key) and whenever the count is greater than half of the array length, return that key(majority element).

- **Algorithm:**

1. Create a hashmap to store a key-value pair, i.e. element-frequency pair.
2. Traverse the array from start to end.
3. For every element in the array, insert the element in the hashmap if the element does not exist as key, else fetch the value of the key ( array[i] ), and increase the value by 1
4. If the count is greater than half then print the majority element and break.
5. If no majority element is found print "No Majority element"

Below is the implementation of above idea:

#### **C++**

```
/* C++ program for finding out majority
```

```
element in an array */
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

**GET \$100 FREE CREDIT**

HIDE AD • AD VIA BUYSPELLADS



```

void findMajority(int arr[], int size)
{
    unordered_map<int, int> m;
    for(int i = 0; i < size; i++)
        m[arr[i]]++;
    int count = 0;
    for(auto i : m)
    {
        if(i.second > size / 2)
        {
            count = 1;
            cout << "Majority found :- " << i.first<<endl;
            break;
        }
    }
    if(count == 0)
        cout << "No Majority element" << endl;
}

// Driver code
int main()
{
    int arr[] = {2, 2, 2, 2, 5, 5, 2, 3, 3};
    int n = sizeof(arr) / sizeof(arr[0]);

    // Function calling
    findMajority(arr, n);

    return 0;
}

// This code is contributed by codeMan_d

```

## Java

```

import java.util.HashMap;

/* Program for finding out majority element in an array */

class MajorityElement

```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```

HashMap<Integer,Integer> map = new HashMap<Integer, Integer>();

for(int i = 0; i < arr.length; i++) {
    if (map.containsKey(arr[i])) {
        int count = map.get(arr[i]) +1;
        if (count > arr.length /2) {
            System.out.println("Majority found :- " + arr[i])
            return;
        } else
            map.put(arr[i], count);
    }
    else
        map.put(arr[i],1);
}
System.out.println(" No Majority element");
}

/* Driver program to test the above functions */
public static void main(String[] args)
{
    int a[] = new int[]{2,2,2,2,5,5,2,3,3};

    findMajority(a);
}
}
// This code is contributed by karan malhotra

```

## Python3

```

# Python3 program for finding out majority
# element in an array

```

```

def findMajority(arr, size):
    m = {}
    for i in range(size):
        if arr[i] in m:
            m[arr[i]] += 1
        else:
            m[arr[i]] = 1

```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
        count = 1
        print("Majority found :-",key)
        break
    if(count == 0):
        print("No Majority element")

# Driver code
arr = [2, 2, 2, 2, 5, 5, 2, 3, 3]
n = len(arr)

# Function calling
findMajority(arr, n)

# This code is contributed by ankush_953
```

## C#

```
// C# Program for finding out majority
// element in an array
using System;
using System.Collections.Generic;

class GFG
{
    private static void findMajority(int[] arr)
    {
        Dictionary<int,
            int> map = new Dictionary<int,
                int>();

        for (int i = 0; i < arr.Length; i++)
        {
            if (map.ContainsKey(arr[i]))
            {
                int count = map[arr[i]] + 1;
                if (count > arr.Length / 2)
                {
                    Console.WriteLine("Majority found :- " +
                                      arr[i]);

                    return;
                }
            }
        }
    }
}
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
        map[arr[i]] = count;
    }

    }
    else
    {
        map[arr[i]] = 1;
    }
}
Console.WriteLine(" No Majority element");
}
```

```
// Driver Code
public static void Main(string[] args)
{
    int[] a = new int[]{2, 2, 2, 2,
                        5, 5, 2, 3, 3};

    findMajority(a);
}
}

// This code is contributed by Shrikant13
```

## Javascript

<script>

// Javascript program for the above approach

```
function findMajority(arr)
{
    let map = new Map();

    for(let i = 0; i < arr.length; i++) {
        if (map.has(arr[i])) {
            let count = map.get(arr[i]) +1;
            if (count > arr.length /2) {
                document.write("Majority found :- " + arr[i]);
                return;
            }
        }
    }
}
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
        }
        else
            map.set(arr[i],1);
        }
        document.write(" No Majority element");
    }

// Driver Code
let a = [ 2,2,2,2,5,5,2,3,3 ];

    findMajority(a);

// This code is contributed by splevel62.
</script>
```

## Output

Majority found :- 2

## Complexity Analysis:

- **Time Complexity:**  $O(n)$ .  
One traversal of the array is needed, so the time complexity is linear.
- **Auxiliary Space:**  $O(n)$ .  
Since a hashmap requires linear space.

Thanks, Ashwani Tanwar, Karan Malhotra for suggesting this.

## METHOD 5

- **Approach:** The idea is to sort the array. Sorting makes similar elements in the array adjacent, so traverse the array and update the count until the present element is similar to the previous one. If the frequency is more than half the size of the array, print the majority element.
- **Algorithm:**



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

2. Traverse the element from start to end.
3. If the current element is equal to the previous element increase the count.
4. Else set the count to 1.
5. If the count is greater than half the size of array, print the element as majority element and break.
6. If no majority element found, print "No majority element"

Below is the implementation of the above idea:

---

## C++

```
// C++ program to find Majority
// element in an array
#include <bits/stdc++.h>
using namespace std;

// Function to find Majority element
// in an array
// it returns -1 if there is no majority element

int majorityElement(int *arr, int n)
{
    if (n == 1) return arr[0];

    int cnt = 1;
    // sort the array, o(nlogn)
    sort(arr, arr + n);
    for (int i = 1; i <= n; i++){
        if (arr[i - 1] == arr[i]){
            cnt++;
        }
        else{
            if (cnt > n / 2){
                return arr[i - 1];
            }
            cnt = 1;
        }
    }
}
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
}

// Driver code
int main()
{
    int arr[] = {1, 1, 2, 1, 3, 5, 1};
    int n = sizeof(arr) / sizeof(arr[0]);

    // Function calling
    cout<<majorityElement(arr, n);

    return 0;
}
```

## Java

```
// Java program to find Majority
// element in an array
import java.io.*;
import java.util.*;

class GFG{

    // Function to find Majority element
    // in an array it returns -1 if there
    // is no majority element
    public static int majorityElement(int[] arr, int n)
    {

        // Sort the array in O(nlogn)
        Arrays.sort(arr);

        int count = 1, max_ele = -1,
            temp = arr[0], ele = 0,
            f = 0;

        for(int i = 1; i <= n; i++)
        {
```



A quick, easy, and intuitive way to build, deploy,  
manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
        if (temp == arr[i])
        {
            count++;
        }
        else
        {
            count = 1;
            temp = arr[i];
        }

        // Sets maximum count and stores
        // maximum occurred element so far
        // if maximum count becomes greater
        // than n/2 it breaks out setting
        // the flag
        if (max_ele < count)
        {
            max_ele = count;
            ele = arr[i];

            if (max_ele > (n / 2))
            {
                f = 1;
                break;
            }
        }
    }

    // Returns maximum occurred element
    // if there is no such element, returns -1
    return (f == 1 ? ele : -1);
}

// Driver code
public static void main(String[] args)
{
    int arr[] = { 1, 1, 2, 1, 3, 5, 1 };
    int n = 7;

    System.out.println(majorityElement(arr, n));
}
}
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

**GET \$100 FREE CREDIT**

HIDE AD • AD VIA BUYSPELLADS



## Python3

```
# Python3 program to find Majority
# element in an array

# Function to find Majority element
# in an array
# it returns -1 if there is no majority element
def majorityElement(arr, n) :

    # sort the array in O(nlogn)
    arr.sort()
    count, max_ele, temp, f = 1, -1, arr[0], 0
    for i in range(1, n) :

        # increases the count if the same element occurs
        # otherwise starts counting new element
        if(temp == arr[i]) :
            count += 1
        else :
            count = 1
            temp = arr[i]

        # sets maximum count
        # and stores maximum occurred element so far
        # if maximum count becomes greater than n/2
        # it breaks out setting the flag
        if(max_ele < count) :
            max_ele = count
            ele = arr[i]

        if(max_ele > (n//2)) :
            f = 1
            break

    # returns maximum occurred element
    # if there is no such element, returns -1
    if f == 1 :
        return ele
    else :
        return -1
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
n = len(arr)

# Function calling
print(majorityElement(arr, n))

# This code is contributed by divyeshrabadiya07
```

## C#

```
// C# program to find Majority
// element in an array
using System;
class GFG
{
    // Function to find Majority element
    // in an array it returns -1 if there
    // is no majority element
    public static int majorityElement(int[] arr, int n)
    {
        // Sort the array in O(nlogn)
        Array.Sort(arr);

        int count = 1, max_ele = -1,
            temp = arr[0], ele = 0,
            f = 0;

        for(int i = 1; i < n; i++)
        {
            // Increases the count if the
            // same element occurs otherwise
            // starts counting new element
            if (temp == arr[i])
            {
                count++;
            }
            else
            {

```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
// Sets maximum count and stores
// maximum occurred element so far
// if maximum count becomes greater
// than n/2 it breaks out setting
// the flag
if (max_ele < count)
{
    max_ele = count;
    ele = arr[i];

    if (max_ele > (n / 2))
    {
        f = 1;
        break;
    }
}

// Returns maximum occurred element
// if there is no such element, returns -1
return (f == 1 ? ele : -1);
}

// Driver code
public static void Main(String[] args)
{
    int []arr = { 1, 1, 2, 1, 3, 5, 1 };
    int n = 7;
    Console.WriteLine(majorityElement(arr, n));
}

// This code is contributed by aashish1995
```

## Javascript

<script>

```
// Javascript program to find Majority
// element in an array
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

```
// is no majority element
function majorityElement(arr, n)
{

    // Sort the array in O(nlogn)
    arr.sort(function(a, b){return a - b});

    let count = 1, max_ele = -1,
        temp = arr[0], ele = 0,
        f = 0;

    for(let i = 1; i < n; i++)
    {

        // Increases the count if the
        // same element occurs otherwise
        // starts counting new element
        if (temp == arr[i])
        {
            count++;
        }
        else
        {
            count = 1;
            temp = arr[i];
        }

        // Sets maximum count and stores
        // maximum occurred element so far
        // if maximum count becomes greater
        // than n/2 it breaks out setting
        // the flag
        if (max_ele < count)
        {
            max_ele = count;
            ele = arr[i];

            if (max_ele > parseInt(n / 2, 10))
            {
                f = 1;
                break;
            }
        }
    }
}
```



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

**GET \$100 FREE CREDIT**

HIDE AD • AD VIA BUYSPELLADS

```
// Returns maximum occurred element
// if there is no such element, returns -1
return (f == 1 ? ele : -1);
}

let arr = [ 1, 1, 2, 1, 3, 5, 1 ];
let n = 7;
document.write(majorityElement(arr, n));

</script>
```

## Output

1

## Complexity Analysis:

- **Time Complexity:**  $O(n \log n)$ .  
Sorting requires  $O(n \log n)$  time complexity.
- **Auxiliary Space:**  $O(1)$ .  
As no extra space is required.

**AMAZON TEST SERIES**  
To Help Crack Your SDE Interview

Enrol Now



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

Like 226

[Previous](#)[Find a triplet that sum to a given value](#)[Next](#)[Two Pointers Technique](#)

## RECOMMENDED ARTICLES

Page : [1](#) [2](#) [3](#)

**01** Check if an array has a majority element  
05, Jan 18

**05** Minimum swaps of same indexed elements required to obtain a Majority Element in one of the arrays  
06, Nov 20

**02** Majority Element | Set-2 (Hashing)  
27, Aug 18

**06** Python3 Program for Check for Majority Element in a sorted array  
11, Nov 21

**03** Majority element in a circular array of 0's and 1's  
14, Sep 18

**07** C Program to Check for Majority Element in a sorted array  
11, Nov 21

**04** Minimum length of subarray in given Ternary Array having 0 as

**08** C++ Program to Check for Majority Element in a sorted



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

[GET \\$100 FREE CREDIT](#)

[HIDE AD](#) • [AD VIA BUYSPELLADS](#)

## Article Contributed By :



GeeksforGeeks

## Vote for difficulty

Current difficulty : [Medium](#)

Easy

Normal

Medium

Hard

Expert

**Improved By :** [tolani62](#), [ukasp](#), [shrikanth13](#), [Akanksha\\_Rai](#), [jit\\_t](#), [codeMan\\_d](#), [ankush\\_953](#), [kokorick](#), [andrew1234](#), [palimohan](#), [shrutikatpara](#), [RohitOberoi](#), [khushboogoyal499](#), [aashish1995](#), [divyeshrabadiya07](#), [avanitrachhadiya2155](#), [rag2127](#), [suresh07](#), [rameshtravel07](#), [mukesh07](#), [saurabh1990aror](#), [splevel62](#), [amartyaghoshgfg](#), [GauravRajput1](#), [boy009](#), [mananbansal1806](#), [hardikkoriintern](#), [srushtirtv18](#)

**Article Tags :** [Accolite](#), [Amazon](#), [D-E-Shaw](#), [FactSet](#), [Majority Element](#), [MakeMyTrip](#), [Microsoft](#), [Moore's Voting Algorithm](#), [Samsung](#), [Arrays](#)

**Practice Tags :** [Accolite](#), [Amazon](#), [Microsoft](#), [Samsung](#), [D-E-Shaw](#), [FactSet](#), [MakeMyTrip](#), [Arrays](#)

Improve Article

Report Issue



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS

Writing code in comment? Please use [ide.geeksforgeeks.org](https://ide.geeksforgeeks.org), generate link and share the link here.

GeeksforGeeks

Load Comments

Sector-136, Noida, Uttar Pradesh - 201305

[feedback@geeksforgeeks.org](mailto:feedback@geeksforgeeks.org)

## Company

About Us  
Careers  
In Media  
Contact Us  
Privacy Policy  
Copyright Policy

## News

Top News  
Technology  
Work & Career  
Business  
Finance  
Lifestyle  
Knowledge

## Web Development

Web Tutorials

## Learn

Algorithms  
Data Structures  
SDE Cheat Sheet  
Machine learning  
CS Subjects  
Video Tutorials  
Courses

## Languages

Python  
Java  
CPP  
Golang  
C#  
SQL  
Kotlin

## Contribute

Write an Article



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSPELLADS



NodeJS

@geeksforgeeks , Some rights reserved

Do Not Sell My Personal Information



A quick, easy, and intuitive way to build, deploy, manage, and scale your apps.

**GET \$100 FREE CREDIT**

HIDE AD • AD VIA BUYSPELLADS