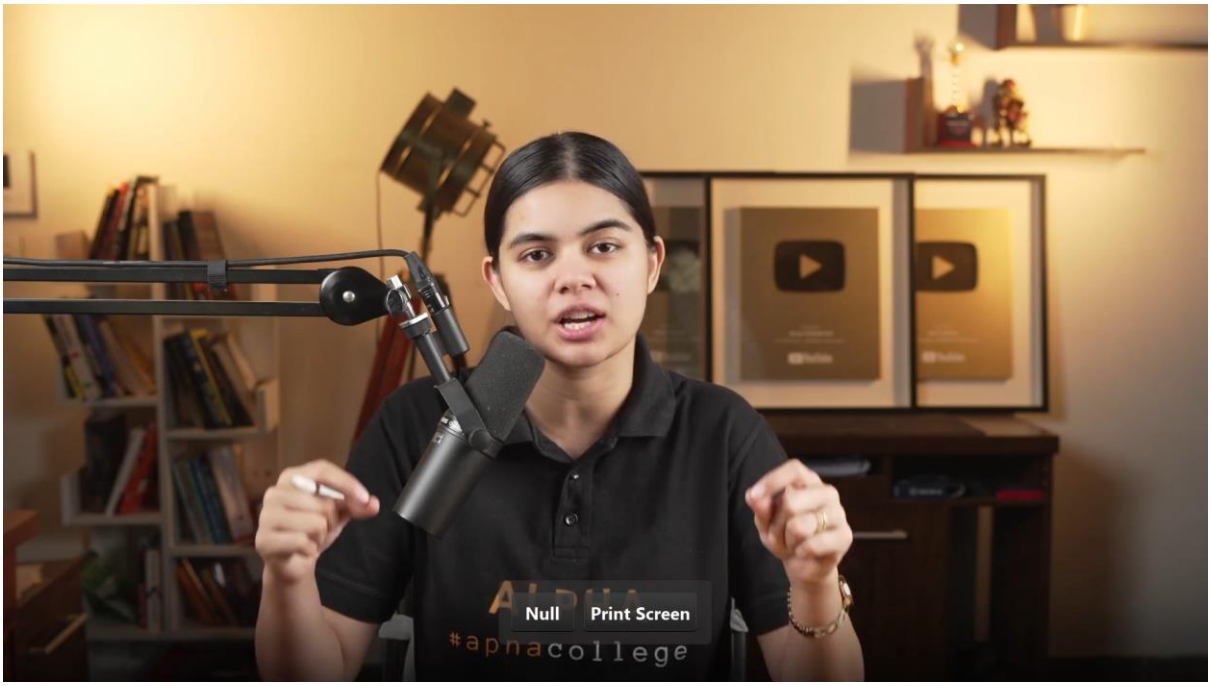
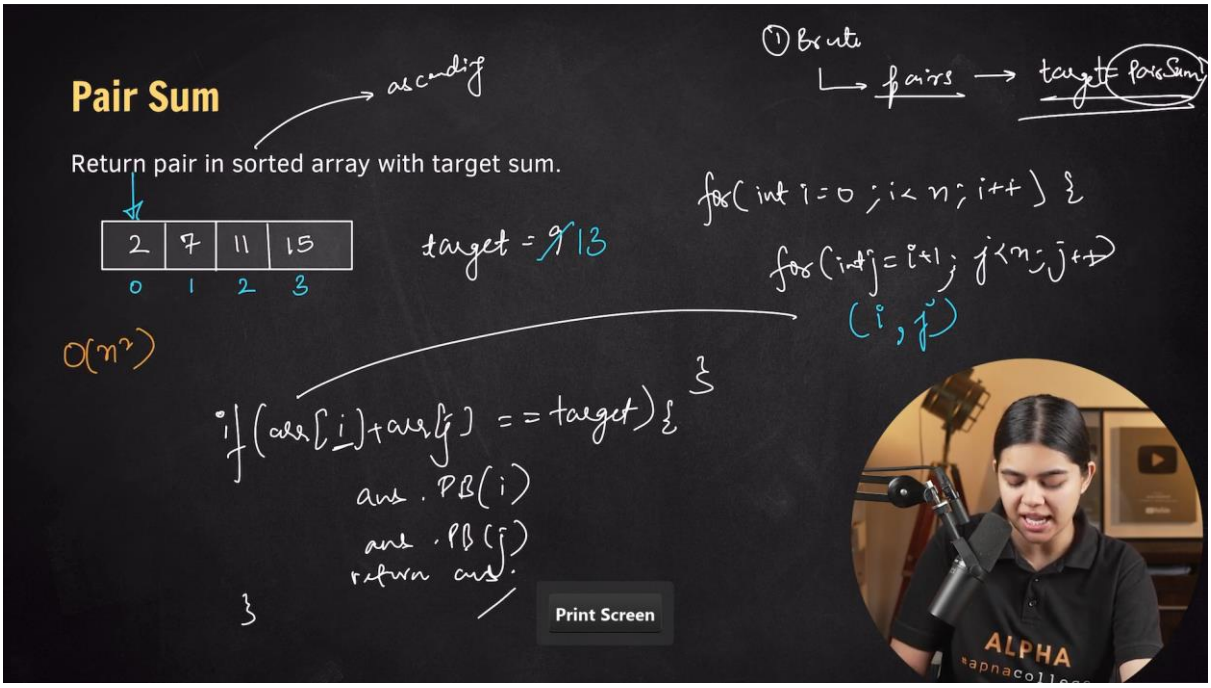
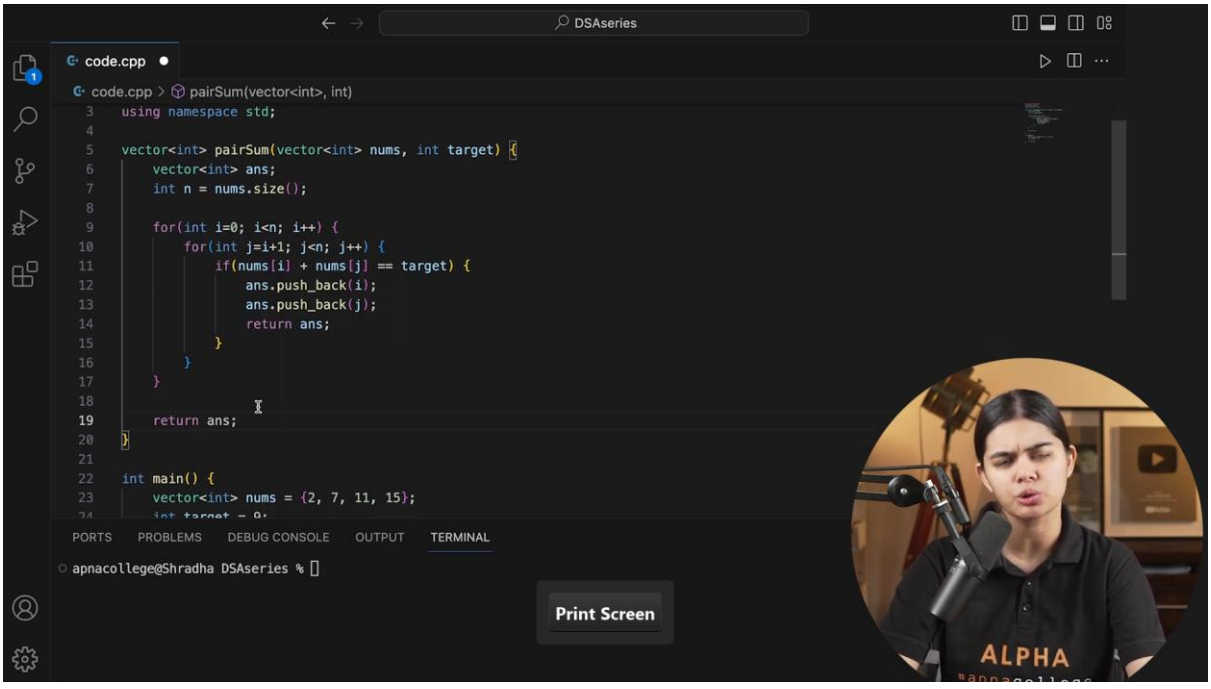

Null
Print Screen
#apnacollege


Pair Sum
ascending
Return pair in sorted array with target sum.
2 7 11 15
0 1 2 3
target = 9 / 13
O(n²)
① Brute
pairs
target = pairSum
for(int i=0; i<n; i++) {
for(int j=i+1; j<n; j++)
(i, j)
if(arr[i]+arr[j] == target) {
ans.PB(i)
ans.PB(j)
return ans;
}
Print Screen
ALPHA


DSAseries
code.cpp
code.cpp > pairSum(vector<int>, int)
using namespace std;
vector<int> pairSum(vector<int> nums, int target) {
vector<int> ans;
int n = nums.size();
for(int i=0; i<n; i++) {
for(int j=i+1; j<n; j++) {
if(nums[i] + nums[j] == target) {
ans.push_back(i);
ans.push_back(j);
return ans;
}
}
}
return ans;
}
int main() {
vector<int> nums = {2, 7, 11, 15};
PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL
apnacollege@Shradha DSAseries %
Print Screen
ALPHA

DSAseries

code.cpp

```cpp
                ans.push_back(i);
                ans.push_back(j);
                return ans;
            }
        }
    }

    return ans;
}

int main() {
    vector<int> nums = {2, 7, 11, 15};
    int target = 13;


    vector<int> ans = pairSum(nums, target);
    cout << ans[0] << ", " << ans[1] << endl;
    return 0;
}
```

PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL

```
apnacollege@Shradha DSAseries % g++ -std=c++11 code.cpp && ./a.out
0, 1
apnacollege@Shradha DSAseries % g++ -std=c++11 code.cpp && ./a.out
0, 2
apnacollege@Shradha DSAseries %
```

Print Screen


## Pair Sum

Return pair in sorted array with target sum.

tar = 26

PS = 26

① Brute → O(n²)

② Optimized

2 pointer approach

| 2 | 7 | 11 | 15 |
|---|---|---|---|
| 0 | 1 | 2 | 3 |

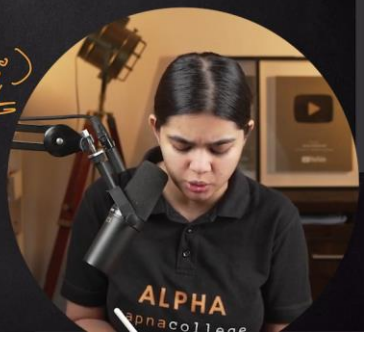
i, j

① PS > tar  j--

② PS < tar  i++

③ PS = tar  ans ⇒ (i, j)

Print Screen




## Pair Sum

Return pair in sorted array with target sum.

tar = 26

PS = 26

O(n)

| 2 | 7 | 11 | 15 |
|---|---|---|---|
| 0 | 1 | 2 | 3 |

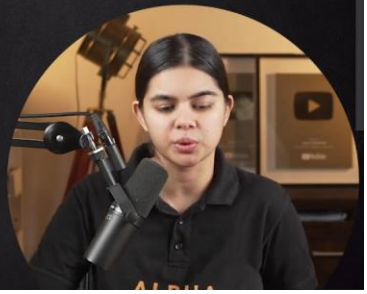
```
i = 0, j = n-1
while (i < j) {
    PS = arr[i] + arr[j]
    if (PS > tar)
        j--
    else if (PS < tar) i++
    else return (i, j)
}
```

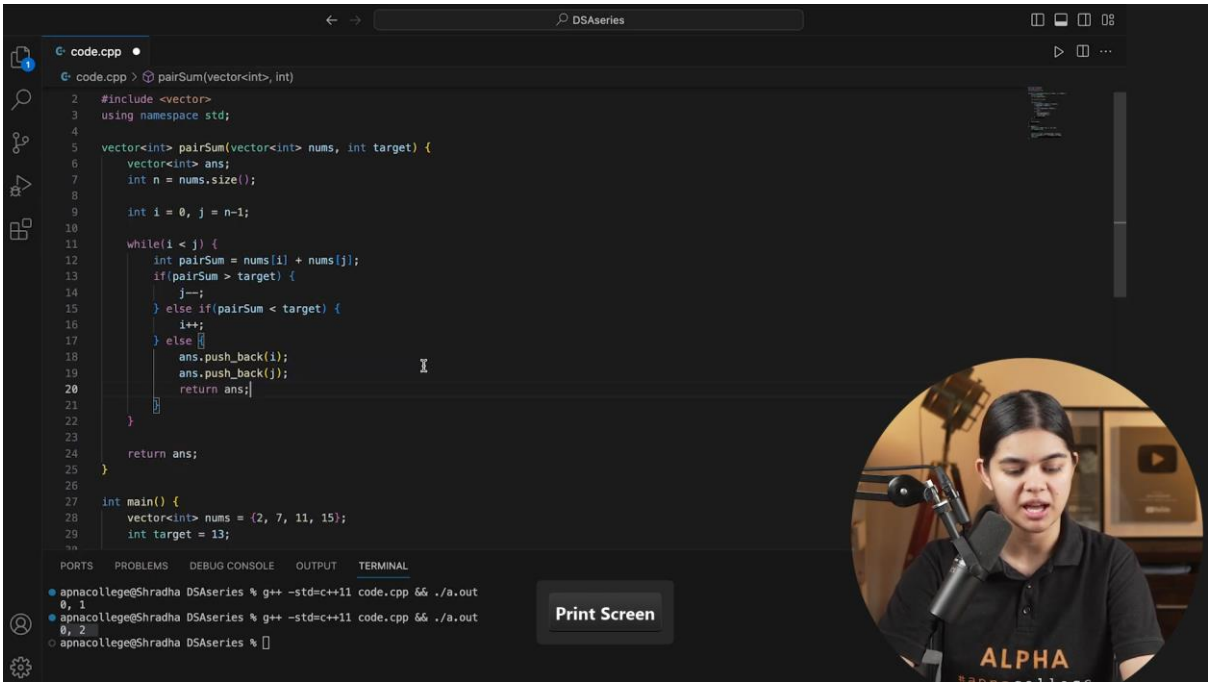
① PS > tar  j--

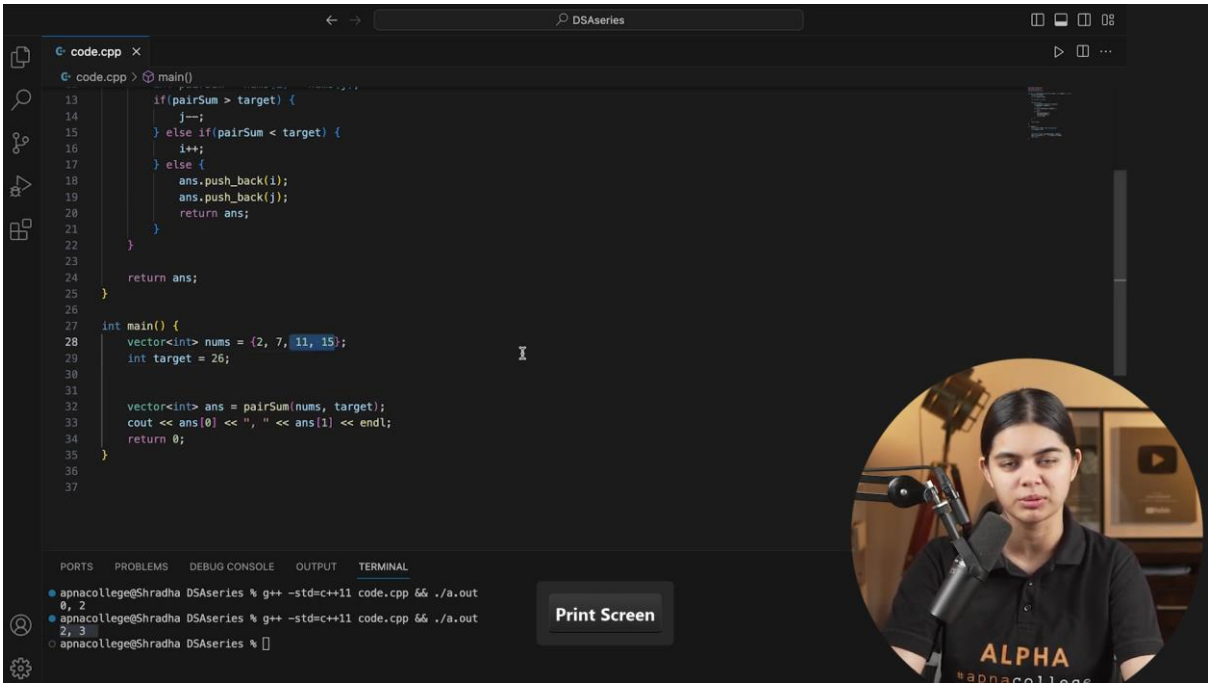
② PS < tar  i++

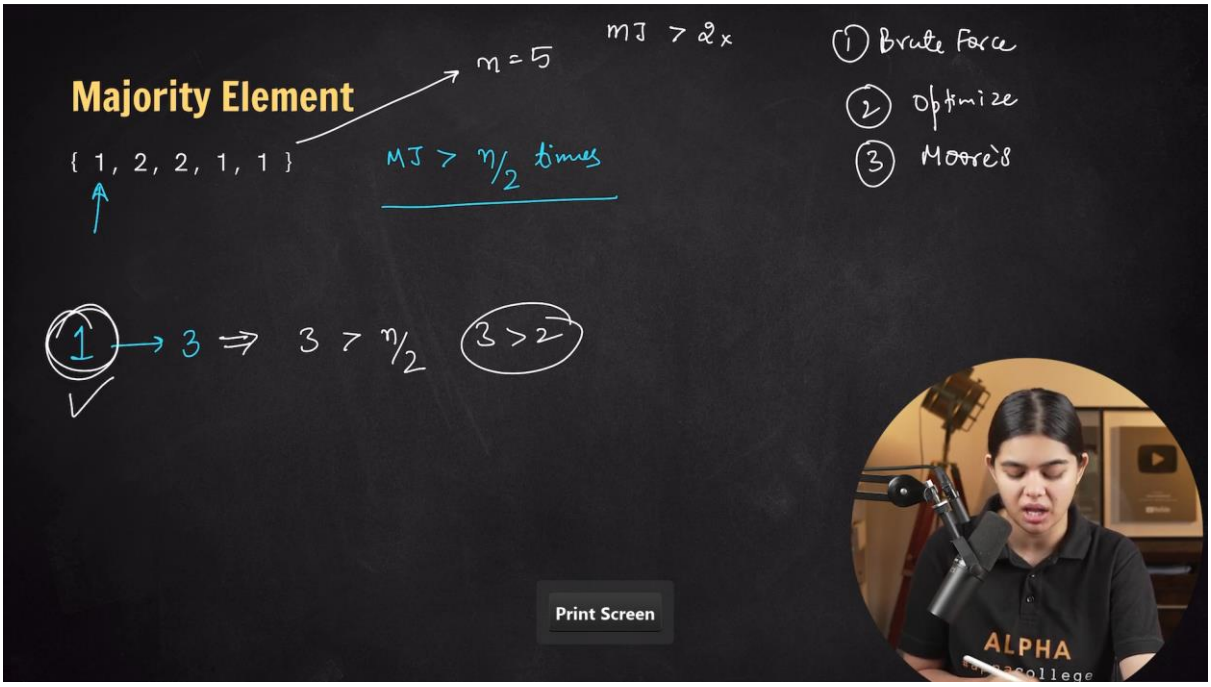
③ PS = tar  ans ⇒ (i, j)

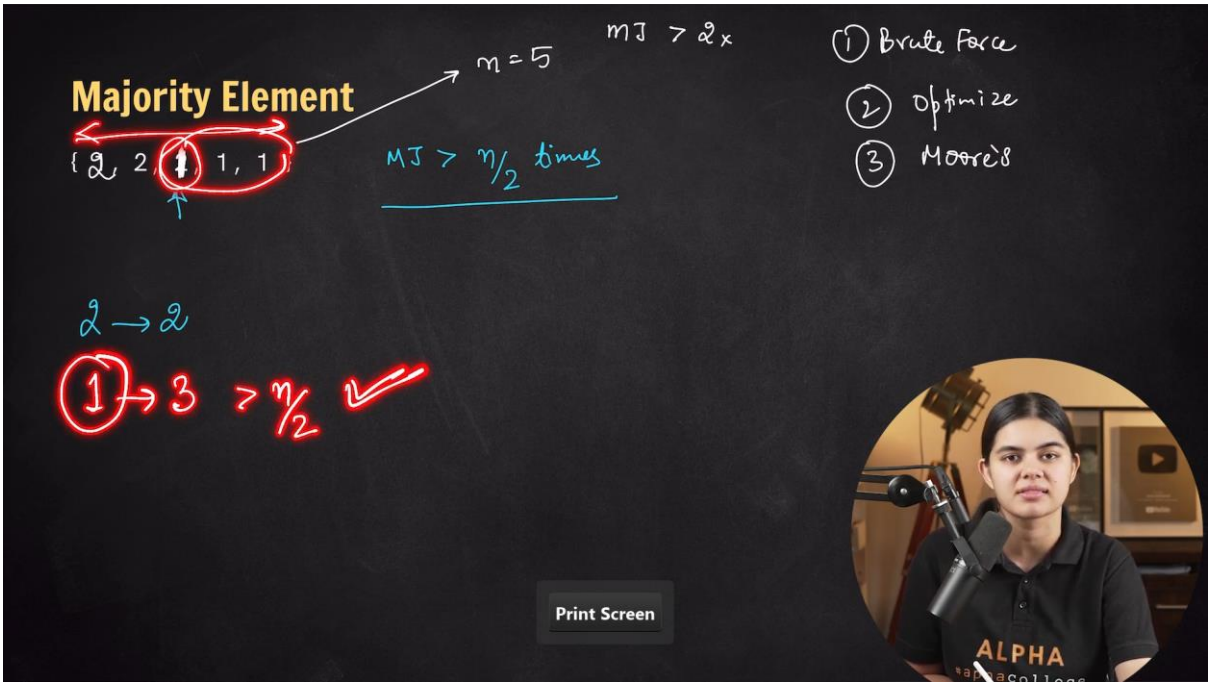
Print Screen

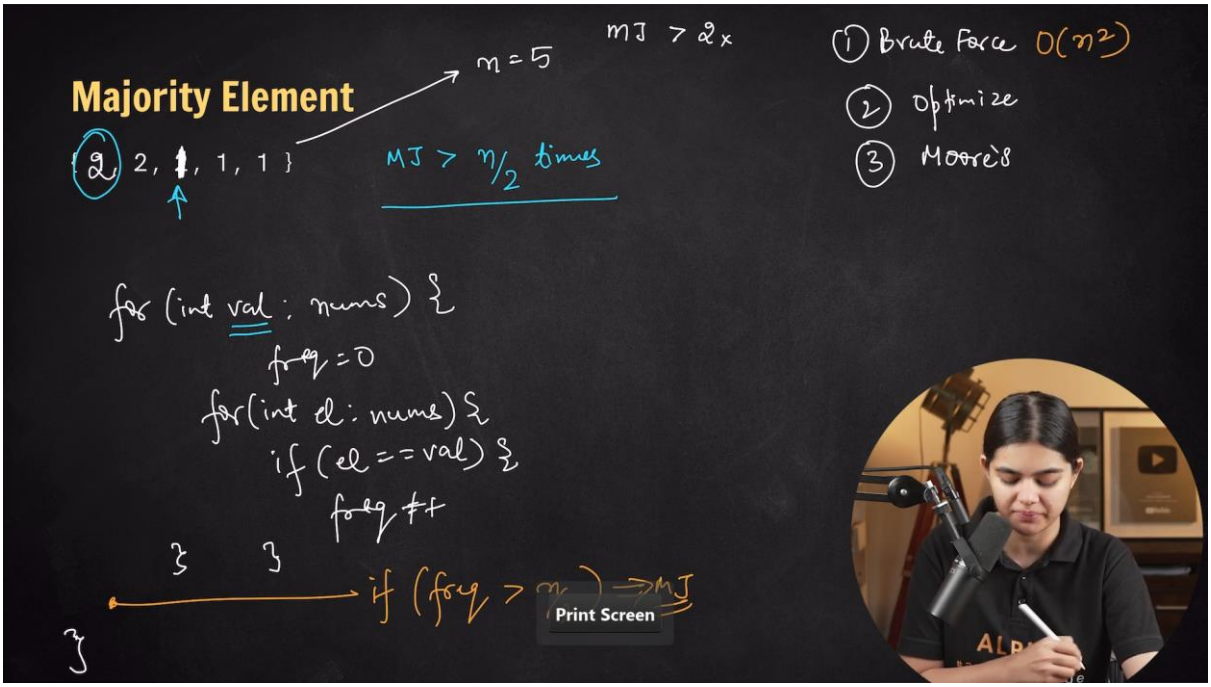


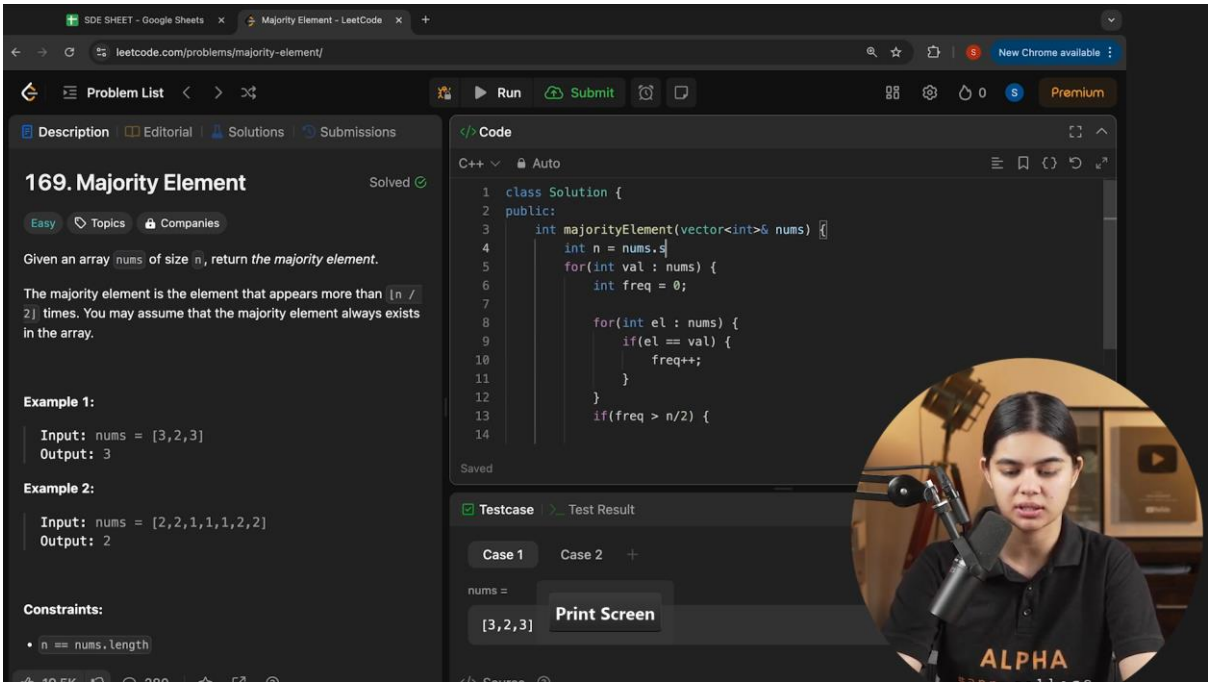

DSAseries
code.cpp
code.cpp > pairSum(vector<int>, int)
#include <vector>
using namespace std;
vector<int> pairSum(vector<int> nums, int target) {
vector<int> ans;
int n = nums.size();
int i = 0, j = n-1;
while(i < j) {
int pairSum = nums[i] + nums[j];
if(pairSum > target) {
j--;
} else if(pairSum < target) {
i++;
} else {
ans.push_back(i);
ans.push_back(j);
return ans;
}
}
return ans;
}
int main() {
vector<int> nums = {2, 7, 11, 15};
int target = 13;
PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL
apnacollege@Shradha DSAseries % g++ -std=c++11 code.cpp && ./a.out
0, 1
apnacollege@Shradha DSAseries % g++ -std=c++11 code.cpp && ./a.out
0, 2
apnacollege@Shradha DSAseries %
Print Screen
ALPHA

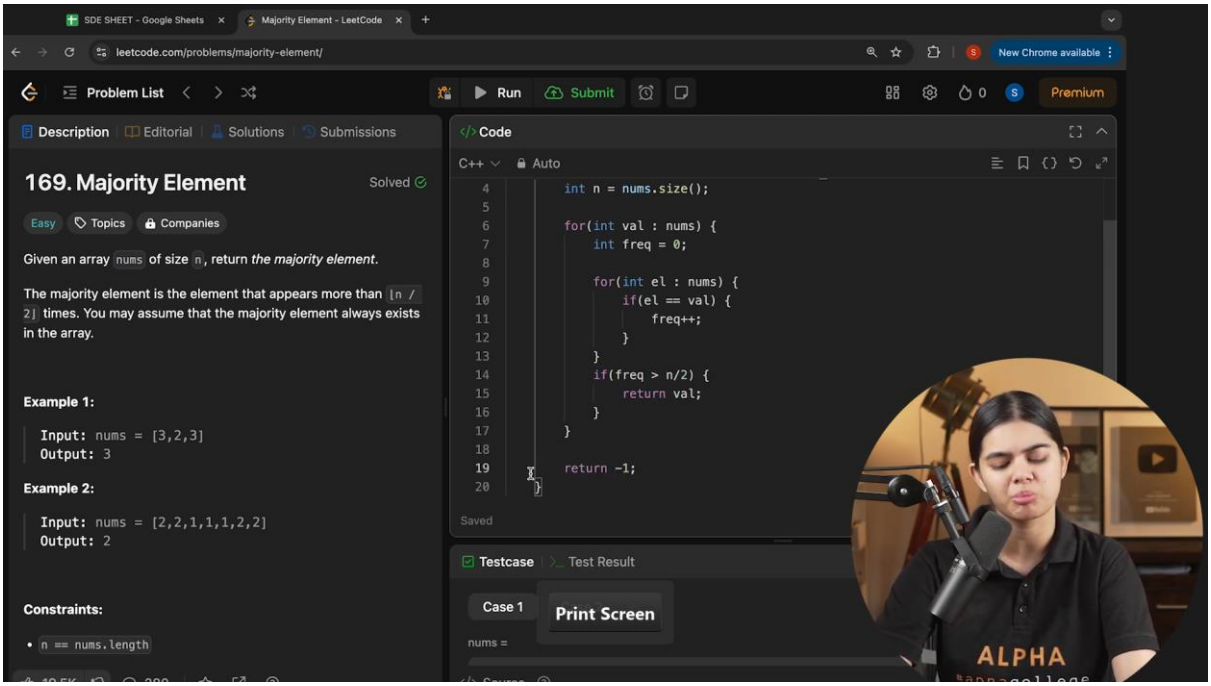

DSAseries
code.cpp
code.cpp > main()
if(pairSum > target) {
j--;
} else if(pairSum < target) {
i++;
} else {
ans.push_back(i);
ans.push_back(j);
return ans;
}
}
return ans;
}
int main() {
vector<int> nums = {2, 7, 11, 15};
int target = 26;
vector<int> ans = pairSum(nums, target);
cout << ans[0] << ", " << ans[1] << endl;
return 0;
}
PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL
apnacollege@Shradha DSAseries % g++ -std=c++11 code.cpp && ./a.out
0, 2
apnacollege@Shradha DSAseries % g++ -std=c++11 code.cpp && ./a.out
2, 3
apnacollege@Shradha DSAseries %
Print Screen
ALPHA

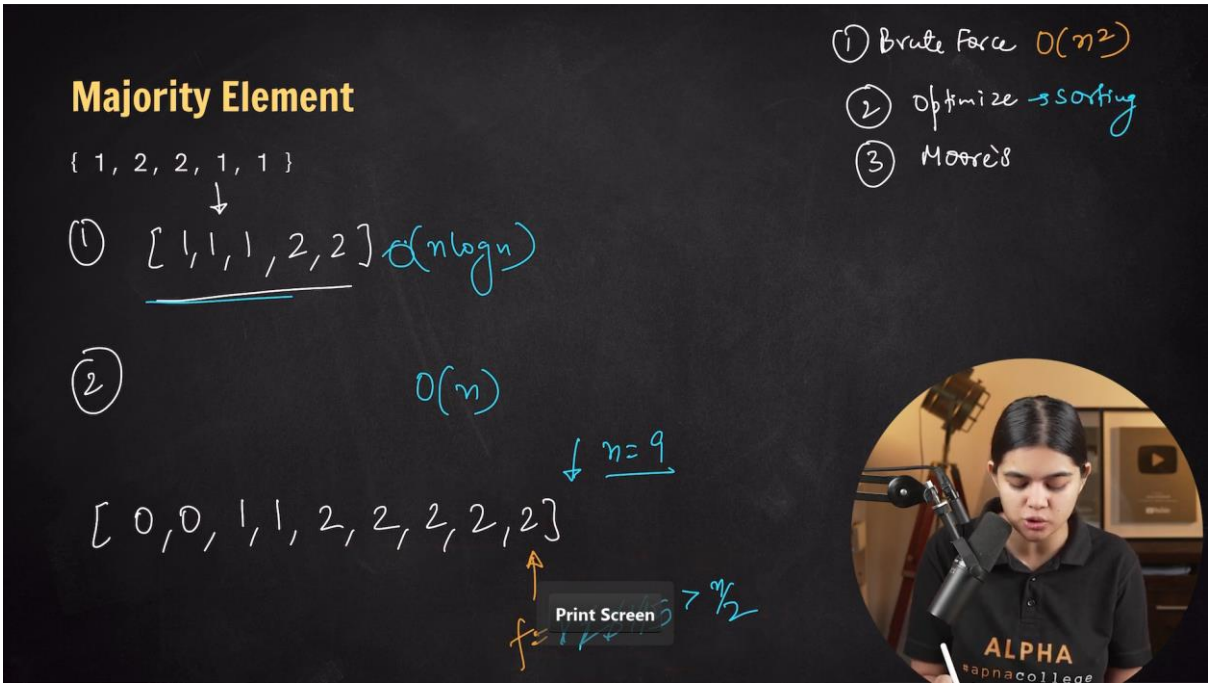

Majority Element
{ 1, 2, 2, 1, 1 }
n = 5
MJ > n/2 times
MJ > 2x
1 Brute Force
2 Optimize
3 Moore's
1 → 3 ⇒ 3 > n/2
3 > 2
Print Screen
ALPHA

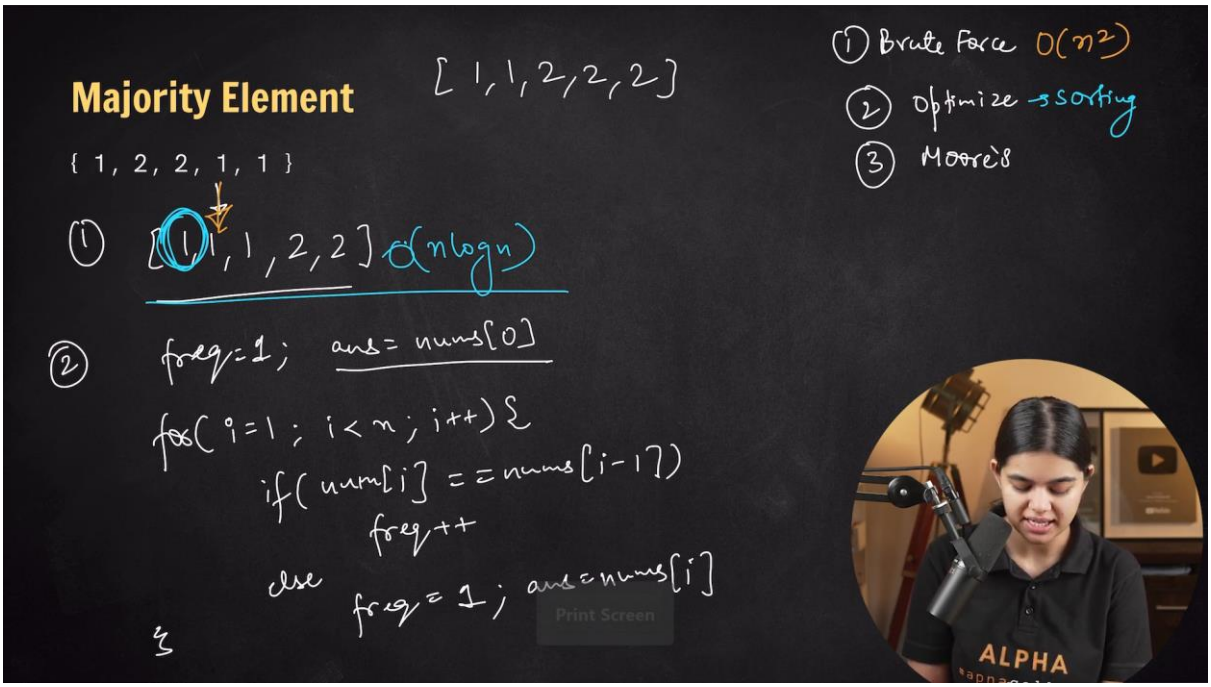

Majority Element
n = 5
MJ > n/2 times
{2, 2, 1, 1, 1}
MJ > 2x
1) Brute Force
2) Optimize
3) Moore's
2 → 2
1 → 3 > n/2
Print Screen
ALPHA

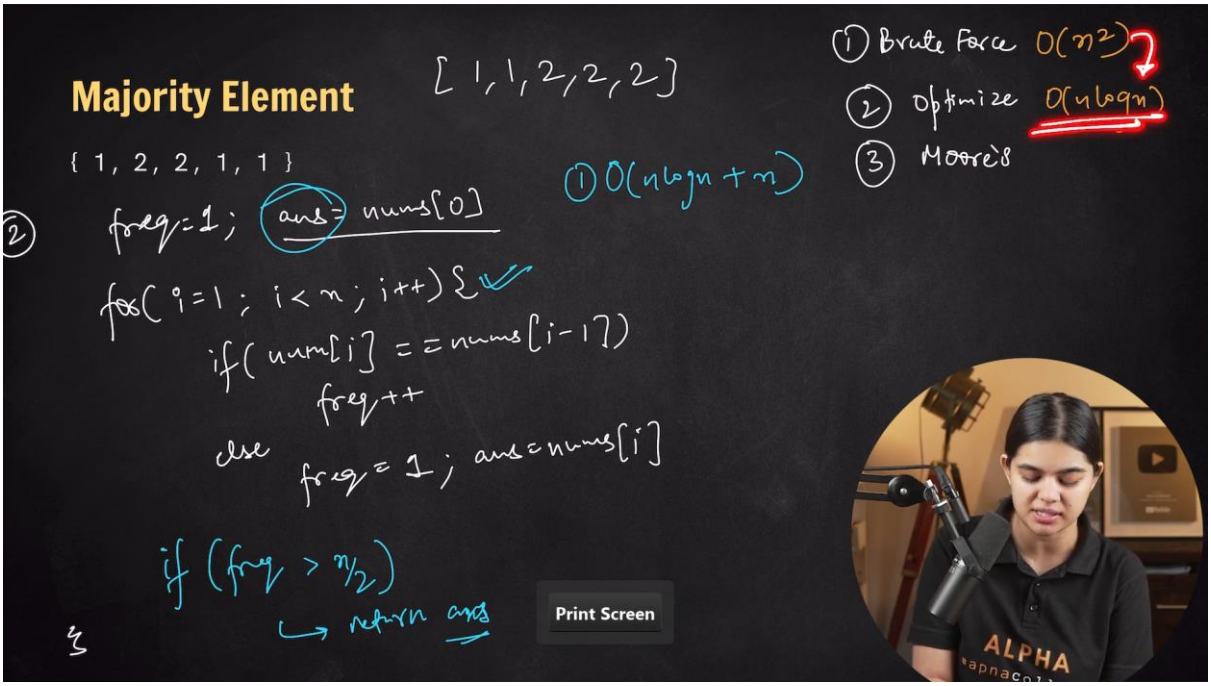

Majority Element
n = 5
MJ > n/2 times
{2, 2, 1, 1, 1}
MJ > 2x
1) Brute Force O(n²)
2) Optimize
3) Moore's
for (int val : nums) {
freq = 0
for(int el : nums) {
if (el == val) {
freq++
}
}
if (freq > n/2) → MJ
}
Print Screen

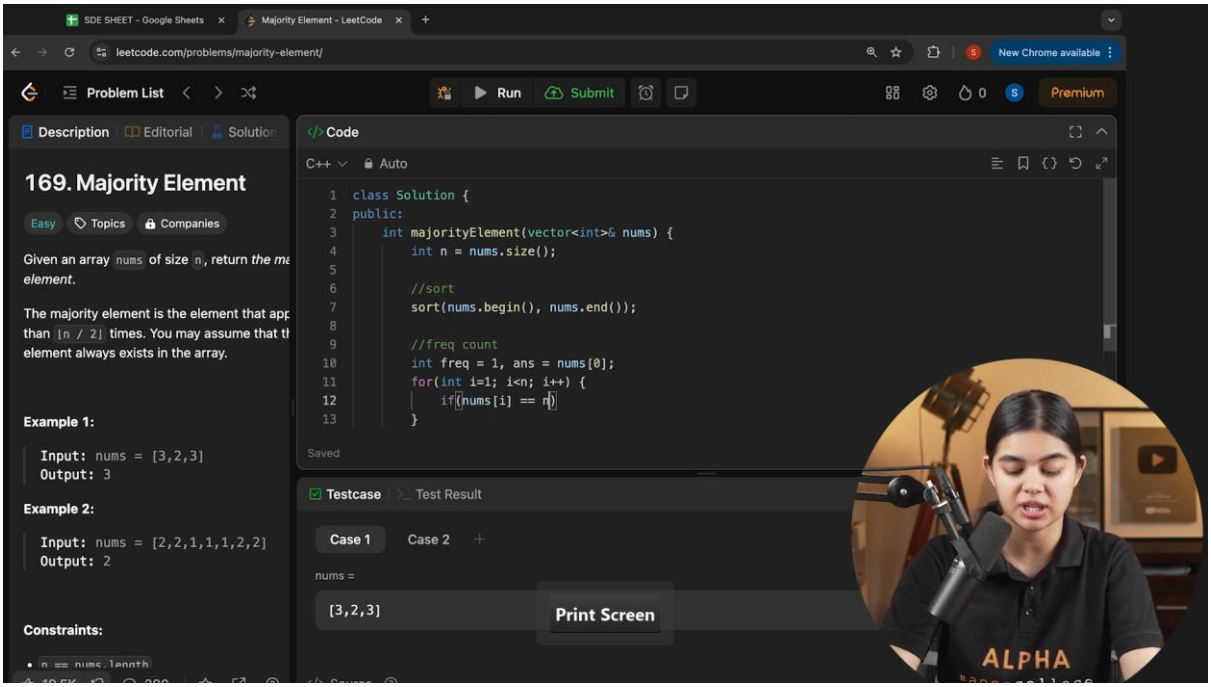

SDE SHEET - Google Sheets
Majority Element - LeetCode
leetcode.com/problems/majority-element/
New Chrome available
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
169. Majority Element
Solved
Easy
Topics
Companies
Given an array nums of size n, return the majority element.
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
Code
C++
Auto
class Solution {
public:
int majorityElement(vector<int>& nums) {
int n = nums.s
for(int val : nums) {
int freq = 0;
for(int el : nums) {
if(el == val) {
freq++;
}
}
if(freq > n/2) {
Saved
Testcase
Test Result
Case 1
Case 2
nums =
[3,2,3]
Print Screen
ALPHA

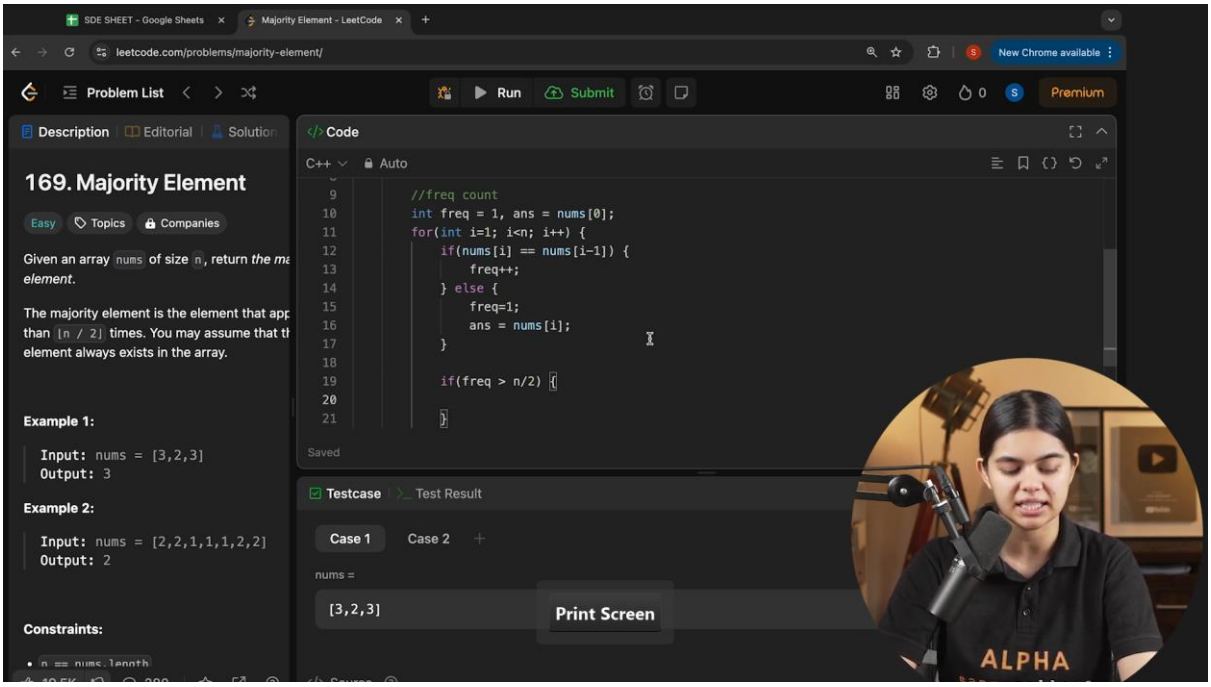

SDE SHEET - Google Sheets
Majority Element - LeetCode
leetcode.com/problems/majority-element/
New Chrome available
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
Code
C++
Auto
169. Majority Element
Solved
Easy
Topics
Companies
Given an array nums of size n, return the majority 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
int n = nums.size();
for(int val : nums) {
int freq = 0;
for(int el : nums) {
if(el == val) {
freq++;
}
}
if(freq > n/2) {
return val;
}
}
return -1;
}
Saved
Testcase
Test Result
Case 1
Print Screen
nums =
ALPHA

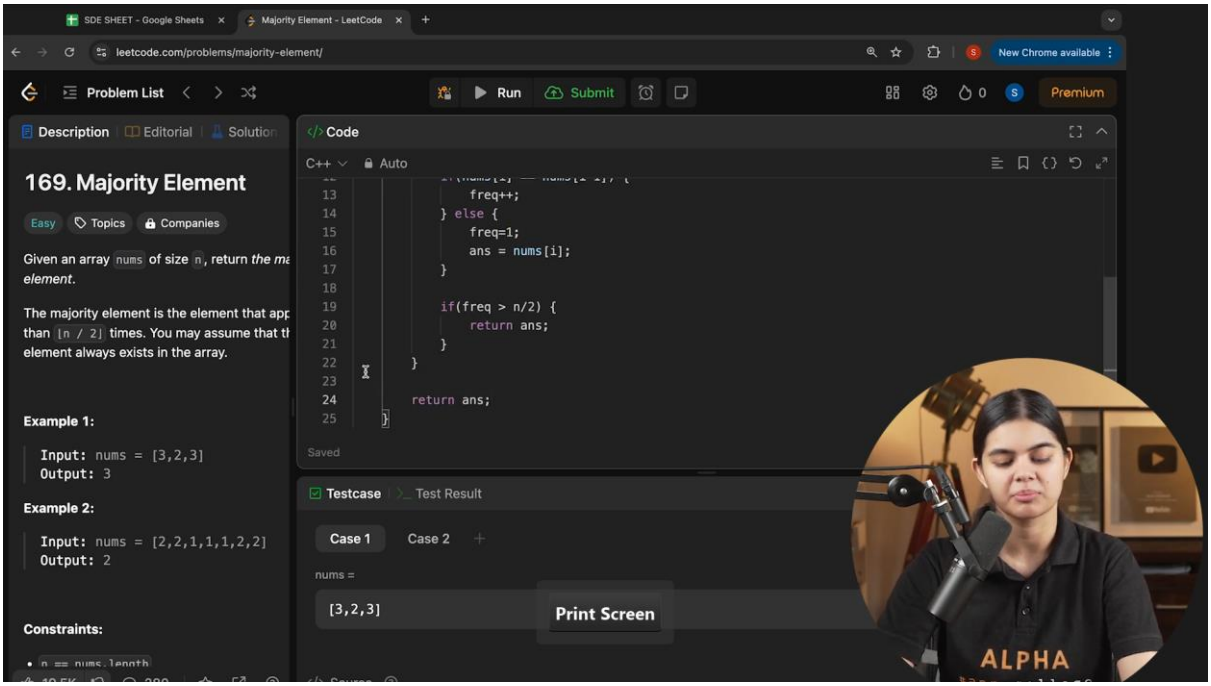

Majority Element
{ 1, 2, 2, 1, 1 }
[ 1, 1, 1, 2, 2 ] O(nlogn)
O(n)
n = 9
[ 0, 0, 1, 1, 2, 2, 2, 2, 2 ]
Print Screen
> n/2
1) Brute Force O(n2)
2) Optimize → sorting
3) Moore's
ALPHA

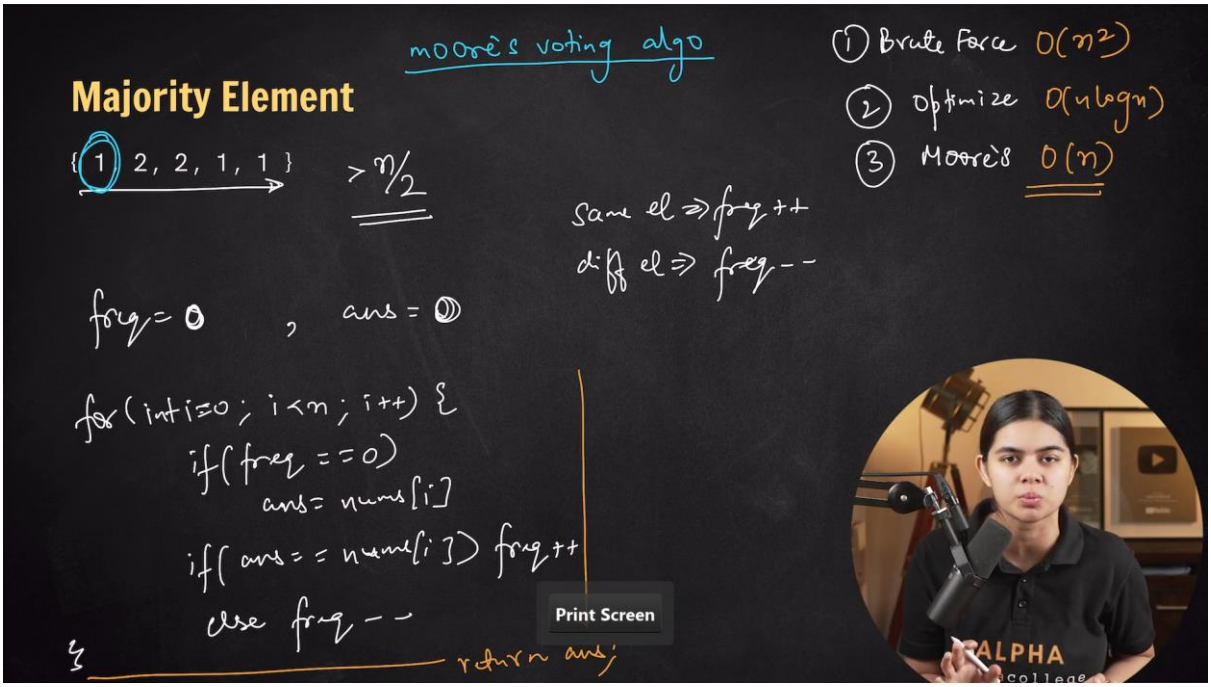

Majority Element
[ 1, 1, 2, 2, 2 ]
{ 1, 2, 2, 1, 1 }
[ 1, 1, 1, 2, 2 ] O(nlogn)
freq = 1; ans = nums[0]
for( i = 1 ; i < n ; i++) {
if( nums[i] == nums[i-1])
freq++
else
freq = 1; ans = nums[i]
}
Print Screen
1) Brute Force O(n2)
2) Optimize → sorting
3) Moore's
ALPHA

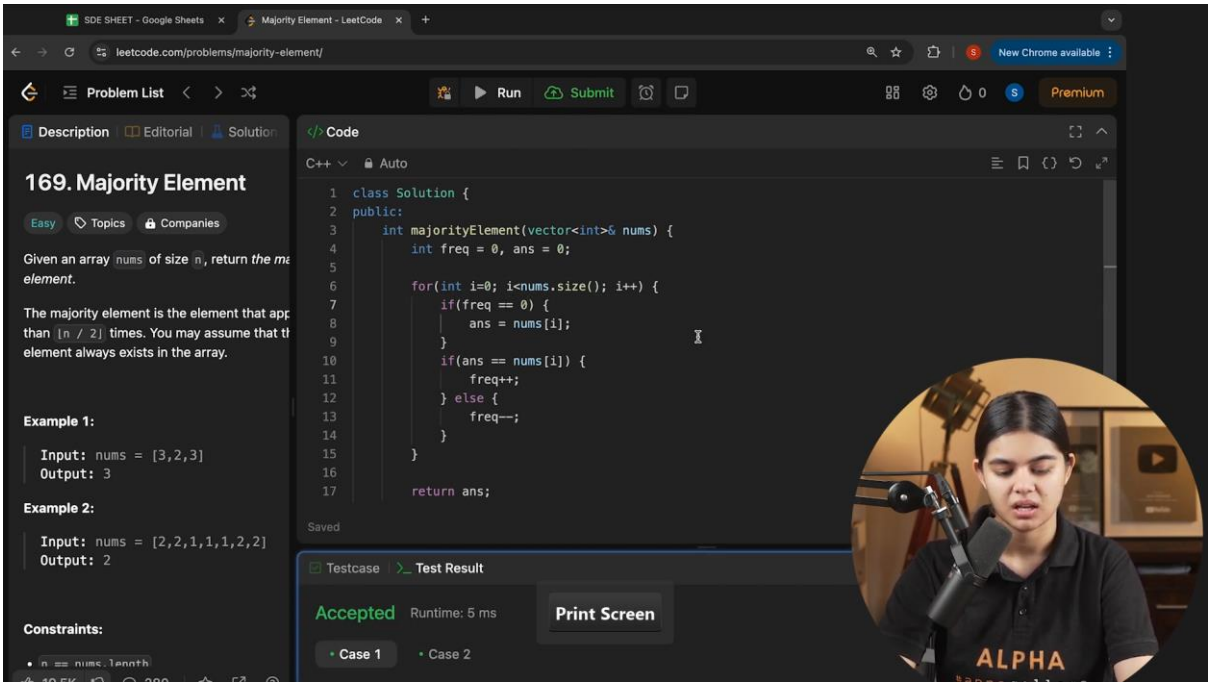

Majority Element
[1,1,2,2,2]
{ 1, 2, 2, 1, 1 }
① Brute Force O(n²)
② Optimize O(nlogn)
③ Moore's
① O(nlogn + n)
freq=1; ans = nums[0]
for(i=1; i<n; i++){
if(nums[i] == nums[i-1])
freq++
else
freq = 1; ans = nums[i]
if (freq > n/2)
return ans
Print Screen
ALPHA


SDE SHEET - Google Sheets
Majority Element - LeetCode
leetcode.com/problems/majority-element/
New Chrome available
Problem List
Run
Submit
Premium
Description
Editorial
Solution
169. Majority Element
Easy
Topics
Companies
Given an array nums of size n, return the ma
element.
The majority element is the element that app
than ⌊n / 2⌋ times. You may assume that th
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
Code
C++
Auto
class Solution {
public:
int majorityElement(vector<int>& nums) {
int n = nums.size();
//sort
sort(nums.begin(), nums.end());
//freq count
int freq = 1, ans = nums[0];
for(int i=1; i<n; i++) {
if(nums[i] == n
}
Saved
Testcase
Test Result
Case 1
Case 2
nums =
[3,2,3]
Print Screen
ALPHA

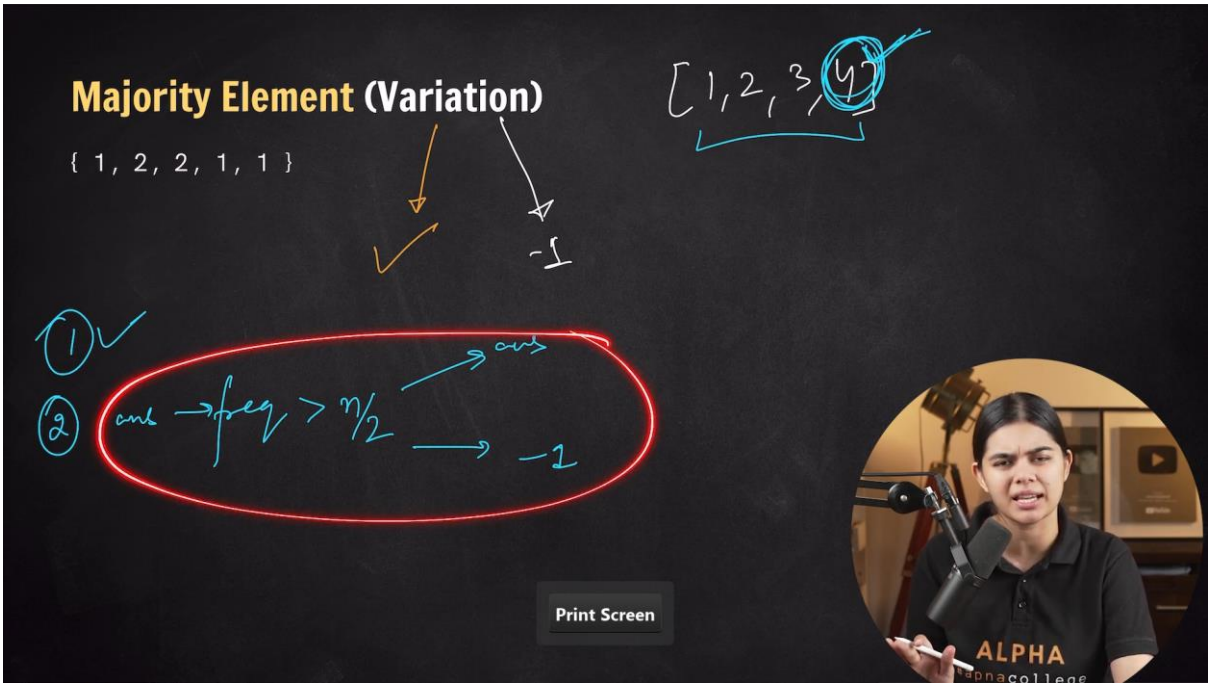

SDE SHEET - Google Sheets
Majority Element - LeetCode
leetcode.com/problems/majority-element/
New Chrome available
Problem List
Run
Submit
Premium
Description
Editorial
Solution
169. Majority Element
Easy
Topics
Companies
Given an array nums of size n, return the ma
element.
The majority element is the element that app
than ⌊n / 2⌋ times. You may assume that th
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
Code
C++
Auto
//freq count
int freq = 1, ans = nums[0];
for(int i=1; i<n; i++) {
if(nums[i] == nums[i-1]) {
freq++;
} else {
freq=1;
ans = nums[i];
}
if(freq > n/2) {
}
Saved
Testcase
Test Result
Case 1
Case 2
nums =
[3,2,3]
Print Screen
ALPHA

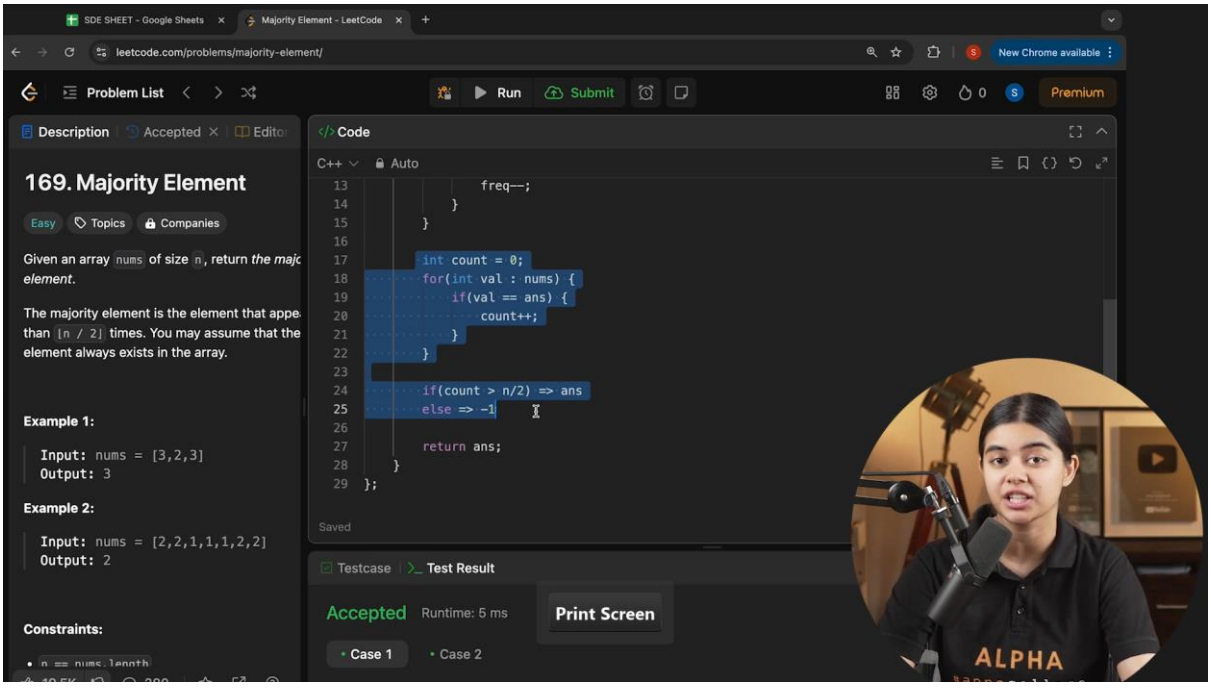

SDE SHEET - Google Sheets
Majority Element - LeetCode
leetcode.com/problems/majority-element/
New Chrome available
Problem List
Run
Submit
Premium
Description
Editorial
Solution
Code
C++
Auto
169. Majority Element
Easy
Topics
Companies
Given an array nums of size n, return the ma
element.
The majority element is the element that app
than ⌊n / 2⌋ times. You may assume that th
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:
freq++;
} else {
freq=1;
ans = nums[i];
}
if(freq > n/2) {
return ans;
}
}
return ans;
}
Saved
Testcase
Test Result
Case 1
Case 2
nums =
[3,2,3]
Print Screen
ALPHA


Majority Element
moore's voting algo
{ 1 2, 2, 1, 1 }
> n/2
1) Brute Force O(n2)
2) Optimize O(nlogn)
3) Moore's O(n)
same el ⇒ freq++
diff el ⇒ freq--
freq = 0 , ans = 0
for(int i=0; i<n; i++) {
if(freq == 0)
ans = nums[i]
if(ans == nums[i]) freq++
else freq--
}
return ans;
Print Screen
ALPHA


SDE SHEET - Google Sheets
Majority Element - LeetCode
leetcode.com/problems/majority-element/
New Chrome available
Problem List
Run
Submit
Premium
Description
Editorial
Solution
Code
C++
Auto
169. Majority Element
Easy
Topics
Companies
Given an array nums of size n, return the ma
element.
The majority element is the element that app
than ⌊n / 2⌋ times. You may assume that th
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:
class Solution {
public:
int majorityElement(vector<int>& nums) {
int freq = 0, ans = 0;
for(int i=0; i<nums.size(); i++) {
if(freq == 0) {
ans = nums[i];
}
if(ans == nums[i]) {
freq++;
} else {
freq--;
}
}
return ans;
}
Saved
Testcase
Test Result
Accepted
Runtime: 5 ms
Print Screen
Case 1
Case 2
ALPHA


Majority Element
{ 1, 2, 2, 1, 1 }
moore's voting algo
(1) Brute Force O(n2)
(2) Optimize O(nlogn)
(3) Moore's O(n)
freq = 0 X 0 X 0 1
ans = 0 1 2 1
Print Screen
ALPHA


Majority Element (Variation)
{ 1, 2, 2, 1, 1 }
[1, 2, 3, 4]
-1
1
2
ans → freq > n/2
ans
-1
Print Screen
ALPHA


SDE SHEET - Google Sheets
Majority Element - LeetCode
leetcode.com/problems/majority-element/
New Chrome available
Problem List
Run
Submit
Premium
Description
Accepted
Editor
169. Majority Element
Easy
Topics
Companies
Given an array nums of size n, return the majo
element.
The majority element is the element that appe
than ⌊n / 2⌋ times. You may assume that the
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
Code
C++
Auto
freq--;
int count = 0;
for(int val : nums) {
if(val == ans) {
count++;
if(count > n/2) => ans
else => -1
return ans;
};
Saved
Testcase
Test Result
Accepted
Runtime: 5 ms
Print Screen
Case 1
Case 2
ALPHA