

[Get 90% Refund](#)[Courses](#)[Tutorials](#)[Practice](#)[Jobs](#)[Problem](#)[Editorial](#)[Submissions](#)[Comments](#)

Output Window

[Compilation Results](#)[Custom Input](#)[Y.O.G.I. \(AI Bot\)](#)

Problem Solved Successfully

[Suggest Feedback](#)

Test Cases Passed

1112 / 1112

Attempts : Correct / Total

1 / 1

Accuracy : 100%

Points Scored

2 / 2

Your Total Score: 50

Time Taken

0.29

Solve Next

[Sorted subsequence of size 3](#)[Array Duplicates](#)[Two Sum - Pair with Given Sum](#)

Stay Ahead With:


Java (21)

[Start Timer](#)

```
1 class Solution {
2     public static int smallestSubwithSum(int x, int[] arr) {
3         int n = arr.length;
4         int currentSum = 0;
5         int minLen = n + 1;
6         int start = 0;
7         for (int end = 0; end < n; end++) {
8             currentSum += arr[end];
9             while (currentSum > x) {
10                 minLen = Math.min(minLen, end - start + 1);
11                 currentSum -= arr[start];
12                 start++;
13             }
14         }
15         return (minLen == n + 1) ? 0 : minLen;
16     }
17 }
```

Ctrl + Enter

[Custom Input](#)[Compile & Run](#)[Submit](#)



Search...

Get 90% Refund

Courses

Tutorials

Practice

Jobs

Problem

Editorial

Submissions

Comments

Output Window

Compilation Results

Custom Input

Y.O.G.I. (AI Bot)

Problem Solved Successfully

Test Cases Passed

1112 / 1112

Attempts : Correct / Total

1 / 1

Accuracy : 100%

Points Scored

2 / 2

Your Total Score: 48

Time Taken

0.81

Solve Next

Bubble Sort

Floor in a Sorted Array

Closest Triplet

Stay Ahead With:

Java (21)

Start Timer


```
1 // User function Template for Java
2 class Solution {
3     public long findMinDiff(ArrayList<Integer> arr, int m) {
4         int n = arr.size();
5         if (m == 0 || n == 0) {
6             return 0;
7         }
8         Collections.sort(arr);
9         if (n < m) {
10            return -1;
11        }
12        long minDiff = Long.MAX_VALUE;
13        for (int i = 0; i <= n - m; i++) {
14            long currentDiff = (long)arr.get(i + m - 1) - arr.get(i);
15
16            if (currentDiff < minDiff) {
17                minDiff = currentDiff;
18            }
19        }
20        return minDiff;
21    }
22 }
```

Ctrl + Enter

Custom Input

Compile & Run

Submit







Get 90% Refund


Courses ▾

Tutorials ▾

Practice ▾

Jobs ▾


 [Problem](#) [Editorial](#) [Submissions](#) [Comments](#)

Output Window

Compilation Results

Custom Input

Y.O.G.I. (AI Bot)

Problem Solved Successfully 

[Suggest Feedback](#)


Test Cases Passed

1111 / 1111


Attempts : Correct / Total

1 / 1

Accuracy : 100%

Points Scored 

4 / 4

Your Total Score: 69 


Time Taken

0.81


Solve Next

[Max sum in the configuration](#) [Boolean Matrix](#) [Row with Minimum 1s](#)


Stay Ahead With:

Java (21) 

```
1 class Solution {
2     public int rowWithMax1s(int arr[][]){
3         int n = arr.length;
4         if (n == 0) return -1;
5         int m = arr[0].length;
6
7         int maxRowIndex = -1;
8
9         int j = m - 1;
10        int i = 0;
11        while (i < n && j >= 0) {
12            if (arr[i][j] == 1) {
13                maxRowIndex = i;
14                j--;
15            }
16            else {
17                i++;
18            }
19        }
20        return maxRowIndex;
21    }
22 }
23 }
```



[Custom Input](#) [Compile & Run](#) [Submit](#)



Search...

Get 90% Refund

Courses

Tutorials

Practice

Jobs

Problem

Editorial

Submissions

Comments

Output Window

Compilation ResultsCustom InputY.O.G.I. (AI Bot)

Problem Solved Successfully

Test Cases Passed

1117 / 1117

Attempts : Correct / Total

1 / 1

Accuracy : 100%

Points Scored

4 / 4

Your Total Score: 65

Time Taken

0.61

Solve Next

Reverse Spiral Form of Matrix

Binary Matrix with at most K 1s

Aggressive Cows

Stay Ahead With:

Java (21)

Start Timer

```
1 class Solution {
2     public int median(int mat[][]){
3         int n = mat.length;
4         int m = mat[0].length;
5
6         int low = 1;
7         int high = 2000;
8
9         int desiredCount = (n * m) / 2;
10
11         while (low <= high) {
12             int mid = low + (high - low) / 2;
13             int count = 0;
14             for (int i = 0; i < n; i++) {
15                 count += countSmallerThanMid(mat[i], mid);
16             }
17             if (count <= desiredCount) {
18                 low = mid + 1;
19             } else {
20                 high = mid - 1;
21             }
22         }
23         return low;
24     }
25     private int countSmallerThanMid(int[] row, int x) {
26         int l = 0, r = row.length - 1;
27         while (l <= r) {
28             int mid = l + (r - l) / 2;
29             if (row[mid] <= x) {
30                 l = mid + 1;
31             } else {
32                 r = mid - 1;
33             }
34         }
35         return l;
36     }
37 }
```

Custom Input

Compile &amp; Run

Submit

Problem List

Submit

Premium

Description

Accepted

Editorial

Solutions

Submissions

All Submissions

Accepted

133 / 133 testcases passed

Editorial

Solution

vi2812 submitted at Feb 13, 2026 01:03

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

43.88 MB | Beats 79.40%

Runtime	1ms	2ms	3ms	4ms
Performance	100%	~0%	~0%	~0%

Code

Java

```
1 class Solution {
2     public boolean searchMatrix(int[][] matrix, int target) {
3         if (matrix == null || matrix.length == 0) return false;
4
5         int m = matrix.length;
6         int n = matrix[0].length;
7
8         int low = 0;
9         int high = (m * n) - 1;
10
11         while (low <= high) {
12             int mid = low + (high - low) / 2;
13
14             int row = mid / n;
15             int col = mid % n;
16             int midValue = matrix[row][col];
17
18             if (midValue == target) {
19                 return true;
20             } else if (midValue < target) {
21                 low = mid + 1;
22             } else {
23                 high = mid - 1;
24             }
25         }
26         return false;
27     }
28 }
```

Testcase

Test Result


Accepted

Runtime: 0 ms

Case 1

Case 2

Input



Get 90% Refund

Courses ▾ Tutorials ▾ Practice ▾ Jobs ▾

⚙️ 🔔 🔔 🔔

V

Problem

Editorial

Submissions

Comments

Output Window

Compilation Results

Custom Input

Y.O.G.I. (AI Bot)

Problem Solved Successfully

Test Cases Passed

1115 / 1115

Attempts : Correct / Total

1 / 1

Accuracy : 100%

Points Scored

4 / 4

Your Total Score: 61

Time Taken

2.07

Suggest Feedback

Solve Next

Find kth element of spiral matrix

Rotate by 90 degree

Reverse Spiral Form of Matrix

Stay Ahead With:

Java (21)

Start Timer

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

```
class Solution {
    public ArrayList<Integer> spirallyTraverse(int[][] mat) {
        ArrayList<Integer> result = new ArrayList<>();
        if (mat == null || mat.length == 0) return result;
        int n = mat.length;
        int m = mat[0].length;
        int top = 0, bottom = n - 1;
        int left = 0, right = m - 1;
        while (top <= bottom && left <= right) {
            for (int j = left; j <= right; j++) {
                result.add(mat[top][j]);
            }
            top++;
            for (int i = top; i <= bottom; i++) {
                result.add(mat[i][right]);
            }
            right--;
            if (top <= bottom) {
                for (int j = right; j >= left; j--) {
                    result.add(mat[bottom][j]);
                }
                bottom--;
            }
            if (left <= right) {
                for (int i = bottom; i >= top; i--) {
                    result.add(mat[i][left]);
                }
                left++;
            }
        }
        return result;
    }
}
```

Ctrl + Enter

Submit



[Get 90% Refund](#)[Courses](#)[Tutorials](#)[Practice](#)[Jobs](#)[Problem](#)[Editorial](#)[Submissions](#)[Comments](#)

Output Window

[Compilation Results](#)[Custom Input](#)[Y.O.G.I. \(AI Bot\)](#)

Problem Solved Successfully

[Suggest Feedback](#)

Test Cases Passed

1115 / 1115

Attempts : Correct / Total

1 / 1

Accuracy : 100%

Points Scored

1 / 1

Your Total Score: 57

Time Taken

0.53

Solve Next

[Multiply Array](#)[Mean of an Array](#)[Greatest of three numbers](#)


Stay Ahead With:

Java (21)

[Start Timer](#)

```
1 class Solution {
2     public double findMedian(int[] arr) {
3         int n = arr.length;
4
5         Arrays.sort(arr);
6
7         if (n % 2 != 0) {
8             return (double) arr[n / 2];
9         } else {
10            int mid1 = arr[n / 2 - 1];
11            int mid2 = arr[n / 2];
12            return (mid1 + mid2) / 2.0;
13        }
14    }
15 }
```

[Custom Input](#)[Compile & Run](#)[Submit](#)



Search...





Get 90% Refund


Courses

Tutorials

Practice

Jobs





Problem

Editorial

Submissions

Comments

Output Window

Compilation Results

Custom Input

Y.O.G.I. (AI Bot)

Problem Solved Successfully

[Suggest Feedback](#)

Test Cases Passed

1112 / 1112

Attempts : Correct / Total

1 / 1

Accuracy : 100%

Points Scored

4 / 4

Your Total Score: 56

Time Taken

0.43

Solve Next

Rearrange Array Alternately

Count Number

Subarray Inversions

Stay Ahead With:

Java (21)

Start Timer



```
1 // User function Template for Java
2
3 class Solution {
4     // Function for finding maximum and value pair
5     int minSwap(int[] arr, int k) {
6         int n = arr.length;
7         int fav = 0;
8         for (int i = 0; i < n; i++) {
9             if (arr[i] <= k) {
10                 fav++;
11             }
12         }
13         if (fav <= 1) return 0;
14         int bad = 0;
15         for (int i = 0; i < fav; i++) {
16             if (arr[i] > k) {
17                 bad++;
18             }
19         }
20         int minSwaps = bad;
21         for (int i = 0, j = fav; j < n; i++, j++) {
22             if (arr[i] > k) {
23                 bad--;
24             }
25             if (arr[j] > k) {
26                 bad++;
27             }
28             minSwaps = Math.min(minSwaps, bad);
29         }
30         return minswaps;
31     }
32 }
```




Custom Input

Compile & Run





Submit





Get 90% Refund

Courses ▾ Tutorials ▾ Practice ▾ Jobs ▾

≡

Problem


Editorial

Submissions

Comments

Output Window

Compilation Results Custom Input Y.O.G.I. (AI Bot)

Problem Solved Successfully 


Test Cases Passed

1111 / 1111

Attempts : Correct / Total

1 / 1


Accuracy : 100%

Points Scored 

2 / 2

Time Taken

0.21

Your Total Score: 52 

Solve Next

Wave Array





Sort by Absolute Difference

Convert an array to reduced form

Stay Ahead With:

Java (21)

Start Timer

```
1 class Solution {
2     // Function to partition the array around the range such
3     // that array is divided into three parts.
4     public void threeWayPartition(int arr[], int a, int b) {
5         int n = arr.length;
6         int low = 0;
7         int mid = 0;
8         int high = n - 1;
9         while (mid <= high) {
10             if (arr[mid] < a) {
11                 int temp = arr[mid];
12                 arr[mid] = arr[low];
13                 arr[low] = temp;
14
15                 low++;
16                 mid++;
17             }
18             else if (arr[mid] > b) {
19                 int temp = arr[mid];
20                 arr[mid] = arr[high];
21                 arr[high] = temp;
22                 high--;
23             }
24             else {
25                 mid++;
26             }
27         }
28     }
29 }
```

Ctrl + Enter

Custom Input Compile & Run Submit