

Problem List

Accepted

Editorial

Solutions

Submissions

Submit

Auto

Premium

Description

Accepted

Editorial

Solutions

Submissions

All Submissions

Accepted 88 / 88 testcases passed

Saurav\_raj01 submitted at Feb 23, 2026 23:59

Editorial

Solution

Runtime

0 ms Beats 100.00%

Analyze Complexity

Memory

48.40 MB Beats 11.90%

Runtime	Beats
0 ms	100.00%
1 ms	0%
2 ms	0%
3 ms	0%
4 ms	0%

Code

Java

```
1 class Solution {
2
3     public int[] searchRange(int[] nums, int target) {
4         int[] ans = new int[2];
5         ans[0] = findFirst(nums, target);
6         ans[1] = findLast(nums, target);
7         return ans;
8     }
9
10    private int findFirst(int[] nums, int target) {
11        int left = 0, right = nums.length - 1;
12        int index = -1;
```

Testcase

Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

nums = [5,7,7,8,8,10]

target = 8

Problem List

Submit

Submit

Ctrl

Enter

code

0

Premium

Description

Accepted

Editorial

Solutions

Submissions

All Submissions

Submit

Ctrl

Enter

code

Accepted

196 / 196 testcases passed

Saurav\_raj01 submitted at Feb 23, 2026 23:58

Editorial

Solution

Runtime

0 ms

Beats 100.00%

Analyze Complexity

Memory

43.80 MB

Beats 82.77%

150%

100%

50%

0%

1ms

2ms

3ms

4ms

Code

Java

```
1 class Solution {
2     public int search(int[] nums, int target) {
3         int left = 0, right = nums.length - 1;
4         while (left <= right) {
5             int mid = left + (right - left) / 2;
6             if (target >= nums[mid] && target < nums[mid + 1]) {
7                 return mid;
8             } else if (target > nums[mid]) {
9                 left = mid + 1;
10            } else {
11                right = mid - 1;
12            }
13        }
14        return -1;
15    }
16 }
```

Testcase

Test Result

Accepted

Runtime: 0 ms

Case 1

Case 2

Case 3

Input

nums =

[4,5,6,7,0,1,2]

target =

0

Output

-1

Array

Submit

0

Premium

Description

Accepted

Editorial

Solutions

Submissions

All Submissions

Accepted

294 / 294 testcases passed

Saurav\_raj01 submitted at Feb 23, 2026 23:56

Editorial

Solution

Runtime

19 ms / Beats 80.59%

Memory

45.83 MB / Beats 63.36%

Analyze Complexity

The graph displays runtime performance over time. The x-axis represents time in milliseconds (1ms to 154ms), and the y-axis represents percentage (0% to 40%). A single sharp peak is visible at 22ms, reaching approximately 35% on the y-axis.

Code

Java

Auto

Ln 12, Col 13

Testcase

Test Result

Accepted

Runtime: 1 ms

Case 1

Case 2

Input

nums =

[1,0,-1,0,-2,2]

target =

0

Code

Java

1 import java.util.\*;

2

3 class Solution {

4 public List<List<Integer>> fourSum(int[] nums, int target) {

5 List<List<Integer>> result = new ArrayList<>();

6 int n = nums.length;

7

8 Arrays.sort(nums);

9

10 for (int i = 0; i < n - 3; i++) {

11

12

Problem List

Submit

Submit

Ctrl

Enter

code

Submit

Ctrl

Enter

code

Description

Accepted

Editorial

Solutions

Submissions

All Submissions

Accepted 88 / 88 testcases passed

Saurav\_raj01 submitted at Feb 23, 2026 23:55

Editorial

Solution

Runtime

131 ms | Beats 71.40%

Analyze Complexity

Memory

43.01 MB | Beats 61.48%

Runtime

3ms 66ms 127ms 180ms 251ms 313ms 375ms 437ms

Code

Java

1 class Solution {  
2 public boolean exist(char[][] board, String word) {  
3 int m = board.length;  
4 int n = board[0].length;  
5  
6 for (int i = 0; i < m; i++) {  
7 for (int j = 0; j < n; j++) {  
8 if (dfs(board, word, i, j, 0)) {  
9 return true;  
10 }  
11 }  
12 }

Java

Auto

1 class Solution {  
2 public boolean exist(char[][] board, String word) {  
3 int m = board.length;  
4 int n = board[0].length;  
5  
6 for (int i = 0; i < m; i++) {  
7 for (int j = 0; j < n; j++) {  
8 if (dfs(board, word, i, j, 0)) {  
9 return true;  
10 }  
11 }  
12 }

Saved

Ln 17, Col 1

Testcase

Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

board =  
[["A", "B", "C", "E"], ["S", "F", "C", "S"], ["A", "D", "E", "E"]]

word =  
"ABCCED"

Array

Submit

Settings

Help

Premium

Description

Accepted

Editorial

Solutions

Submissions

All Submissions

Accepted 10 / 10 testcases passed

Saurav\_raj01 submitted at Feb 23, 2026 23:52

Editorial

Solution

Runtime

1 ms | Beats 87.91%

Analyze Complexity

Memory

44.12 MB | Beats 76.20%

Analyze Complexity

Time (ms)	Percentage
1ms	87.91%
2ms	0%
3ms	0%
4ms	0%

Code

Java

Auto

```
1 import java.util.*;
2
3 class Solution {
4     public List<List<Integer>> subsets(int[] nums) {
5         List<List<Integer>> result = new ArrayList<>();
6         int n = nums.length;
7         int total = 1 << n; // 2^n
8
9         for (int mask = 0; mask < total; mask++) {
10             List<Integer> subset = new ArrayList<>();
11             for (int i = 0; i < n; i++) {
12
```

Testcase

Test Result

Accepted

Runtime: 0 ms

Case 1

Case 2

Input

nums = [1,2,3]

Output

[[], [1], [2], [1,2], [3], [1,3], [2,3], [1,2,3]]

Problem List

Submit

Auto

Premium

Description

Accepted

Editorial

Solutions

Submissions

All Submissions

Accepted 89 / 89 testcases passed

Saurav\_raj01 submitted at Feb 23, 2026 23:51

Editorial

Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

43.63 MB | Beats 42.16%

Runtime	Beats
0 ms	100.00%
1 ms	~0%
2 ms	~0%
3 ms	~0%
4 ms	~0%

Code

Java

Auto

```
1 class Solution {
2     public void sortColors(int[] nums) {
3         int low = 0, mid = 0, high = nums.length - 1;
4
5         while (mid <= high) {
6             if (nums[mid] == 0) {
7                 int temp = nums[low];
8                 nums[low] = nums[mid];
9                 nums[mid] = temp;
10                low++;
11                mid++;
12            }
13        }
14    }
15 }
```

Saved

Ln 16, Col 20

Testcase

Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

nums =

[2,0,2,1,1,0]

Output

[0,0,1,1,2,2]

Code

Java

```
1 class Solution {
2     public void sortColors(int[] nums) {
3         int low = 0, mid = 0, high = nums.length - 1;
4
5         while (mid <= high) {
```

Problem List

Submit

Submit

Premium

Description

Accepted

Editorial

Solutions

Submissions

All Submissions

Accepted 133 / 133 testcases passed

Saurav\_raj01 submitted at Feb 23, 2026 23:49

Editorial

Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

43.75 MB | Beats 91.32%

Runtime	Beats
0 ms	100.00%
1 ms	~0%
2 ms	~0%
3 ms	~0%
4 ms	~0%

Code

Java

Auto

```
int m = matrix.length;
int n = matrix[0].length;

int left = 0;
int right = m * n - 1;

while (left <= right) {
    int mid = left + (right - left) / 2;

    int val = matrix[mid / n][mid % n];

    if (val == target) return true;
}
```

Testcase

Test Result

Accepted

Runtime: 0 ms

Case 1

Case 2

Input

matrix =

[[1,3,5,7], [10,11,16,20], [23,30,34,60]]

target =

3

Problem List

Submit

Settings

0

Profile

Premium

Description

Accepted

Editorial

Solutions

Submissions

All Submissions

Accepted

202 / 202 testcases passed

Saurav\_raj01 submitted at Feb 23, 2026 23:47

Editorial

Solution

Runtime

1 ms | Beats 60.39%

Analyze Complexity

Memory

47.71 MB | Beats 13.78%

Runtime (ms)	Beats (%)
1ms	60.39%
2ms	13.78%
3ms	13.78%
4ms	13.78%

Code

Java

Auto

```
1 class Solution {
2     public void setZeroes(int[][] matrix) {
3         int m = matrix.length;
4         int n = matrix[0].length;
5
6         boolean firstRowZero = false;
7         boolean firstColZero = false;
8
9         for (int j = 0; j < n; j++) {
10             if (matrix[0][j] == 0) {
11                 firstRowZero = true;
12                 break;
13             }
14         }
15     }
16 }
```

Saved

Ln 8, Col 9

Testcase

Test Result

Accepted

Runtime: 0 ms

Case 1

Case 2

Input

matrix =

[[1,1,1],[1,0,1],[1,1,1]]

Output

[[1,0,1],[0,0,0],[1,0,1]]

Problem List

Accepted

Editorial

Solutions

Submissions

Run Ctrl

Submit

0

Premium

All Submissions

Accepted 114 / 114 testcases passed

Saurav\_raj01 submitted at Feb 23, 2026 23:45

Editorial

Solution


Runtime

0 ms Beats 100.00%

Analyze Complexity

Memory

43.52 MB Beats 38.32%



Code | Java

```
1 class Solution {
2     public int[] plusOne(int[] digits) {
3         int n = digits.length;
4
5         for (int i = n - 1; i >= 0; i--) {
```

Code

Java

Auto

```
1 class Solution {
2     public int[] plusOne(int[] digits) {
3         int n = digits.length;
4
5         for (int i = n - 1; i >= 0; i--) {
6             if (digits[i] < 9) {
7                 digits[i]++;
8                 return digits;
9             }
10            digits[i] = 0;
11        }
12    }
```

Saved

Ln 18, Col 1

Testcase

Test Result

Accepted Runtime: 0 ms

Case 1

Case 2

Case 3

Input

digits =

[1,2,3]

Output

[1,2,4]

Problem List

Submit

Premium

Description

Accepted

Editorial

Solutions

Submissions

All Submissions

Accepted 128 / 128 testcases passed

Saurav\_raj01 submitted at Feb 23, 2026 23:44

Editorial

Solution

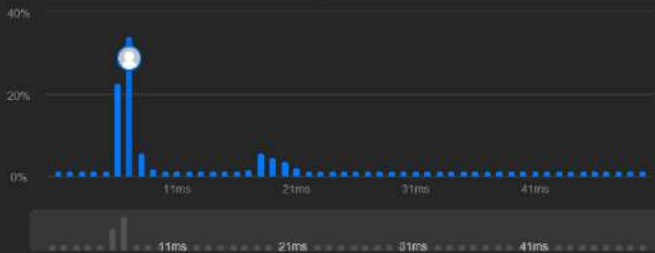
Runtime

7 ms | Beats 76.62%

Analyze Complexity

Memory

49.64 MB | Beats 75.18%



Code

Java

Auto

15

16

17

18

19

20

21

22

23

24

25

26

Saved

Ln 26, Col 1

Testcase

Test Result

Accepted Runtime: 0 ms

Case 1

Case 2

Case 3

Input

strs =

["eat","tea","tan","ate","nat","bat"]

Output

[["eat","tea","ate"],["bat"],["tan","nat"]]

Code

Java

1

2

3

4

5

import java.util.\*;

class Solution {

public List<List<String>> groupAnagrams(String[] strs) {

Problem List

Description

Accepted

Editorial

Solutions

Submissions

All Submissions

Accepted 110 / 110 testcases passed

Saurav\_raj01 submitted at Feb 23, 2026 23:43

Editorial

Solution

Runtime

1 ms Beats 99.60%

Analyze Complexity

Memory

47.42 MB Beats 36.30%

100%

50%

0%

17ms 33ms 50ms 66ms 83ms 99ms 116ms

Code

Java

```
1 class Solution {
2     public int jump(int[] nums) {
3         int jumps = 0;
4         int end = 0;
5         int far = 0;
6         for (int i = 0; i < nums.length; i++) {
7             far = Math.max(far, i + nums[i]);
8             if (i == end) {
9                 jumps++;
10                end = far;
11            }
12        }
13        return jumps;
14    }
15 }
```

Saved In 10, Col 30

Testcase

Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

nums = [2,3,1,1,4]

Output

2

Problem List

Submit

Settings

0

Power

Profile

Premium

Description

Accepted

Editorial

Solutions

Submissions

All Submissions

Accepted

176 / 176 testcases passed

Saurav\_raj01 submitted at Feb 23, 2026 23:42

Editorial

Solution

Runtime

6 ms | Beats 73.38%

Analyze Complexity

Memory

47.46 MB | Beats 11.25%

Time Range	Percentage
1ms	~2%
2ms	~2%
3ms	~2%
4ms	~2%
5ms	~25%
6ms	~35%
7ms	~25%

Code

Java

Auto

```
26
27
28
29
30
31
32
33
    backtrack(arr, target - arr[i], i + 1, temp, result);
    temp.remove(temp.size() - 1);
}
```

SavedLn 20, Col 1

Testcase

Test Result

Accepted

Runtime: 1 ms

Case 1

Case 2

Input

candidates =

[10,1,2,7,6,1,5]

target =

8

Code

Java

```
1 import java.util.*;
2
3 class Solution {
4     public List<List<Integer>> combinationSum2(int[] candidates, int target)
5         List<List<Integer>> result = new ArrayList<>();
```

Problem List

Submit

0

Premium

Description

Accepted

Editorial

Solutions

Submissions

All Submissions

Accepted 160 / 160 testcases passed

Saurev\_rag01 submitted at Feb 23, 2026 23:38

Editorial

Solution

Runtime

2 ms | Beats 95.12%

Analyze Complexity

Memory

46.04 MB | Beats 19.62%

Time Interval	Percentage
1ms	~5%
2ms	~65%
3ms	~25%
4ms	~5%
5ms	~5%

Code

Java

```
1 import java.util.*;
2
3 class Solution {
4     public List<List<Integer>> combinationSum(int[] candidates, int target)
5         List<List<Integer>> result = new ArrayList<>();
```

Code

Auto

```
20 for (int i = start; i < candidates.length; i++) {
21     temp.add(candidates[i]);
22     backtrack(candidates, target - candidates[i], i, temp, result);
23     temp.remove(temp.size() - 1);
24 }
25 }
26 }
27 }
```

Testcase

Test Result

Accepted

Runtime: 0 ms

Case 1

Case 2

Case 3

Input

candidates =

[2,3,6,7]

target =

7

Array

Submit

0

Premium

Description

Accepted

Editorial

Solutions

Submissions

All Submissions

Accepted

66 / 66 testcases passed

Saurav\_raj01 submitted at Feb 23, 2026 23:37

Editorial

Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

44.96 MB | Beats 27.16%

Runtime	Beats
0 ms	100.00%
1 ms	0%
2 ms	0%
3 ms	0%
4 ms	0%

Code

Java

Auto

right = mid - 1;

}

}

return left;

}

1

23

Saved

Ln 1, Col 1

Testcase

Test Result

Accepted

Runtime: 0 ms

Case 1

Case 2

Case 3

Input

nums =

[1,3,5,6]

target =

5

Code

Java

1 class Solution {

2 public int searchInsert(int[] nums, int target) {

3 int left = 0;

4 int right = nums.length - 1;

5