

Problem List < > ✎

Description Accepted Editorial Solutions Submissions

Accepted 88 / 88 testcases passed Saurav_raj01 submitted at Feb 23, 2026 23:59

Runtime 0 ms Beats 100.00% Memory 48.40 MB Beats 11.90%

Analyze Complexity

100% 50% 0% 1ms 2ms 3ms 4ms

Code Java

```
1 class Solution {  
2     public int[] searchRange(int[] nums, int target) {  
3         int[] ans = new int[2];  
4         ans[0] = findFirst(nums, target);  
5         ans[1] = findLast(nums, target);  
6         return ans;  
7     }  
8  
9     private int findFirst(int[] nums, int target) {  
10        int left = 0, right = nums.length - 1;  
11        int index = -1;  
12    }  
13}
```

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 < > ✎

Description Accepted Editorial Solutions Submissions Submit Ctrl Enter Code Premium

All Submissions

Accepted 196 / 196 testcases passed
Saurav_raj01 submitted at Feb 23, 2026 23:58

Runtime: 0 ms | Beats 100.00% Memory: 43.80 MB | Beats 82.77%

Analyze Complexity

100%
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             if (target >= nums[left] && target < nums[mid]) {
6                 right = mid - 1;
7             } else {
8                 left = mid + 1;
9             }
10            else {
11                if (target > nums[mid] && target <= nums[right]) {
12                    left = mid + 1;
13                } else {
14                    right = mid - 1;
15                }
16            }
17        }
18    }
19}
```

Testcase: Test Result

Accepted Runtime: 0 ms:
Case 1 Case 2 Case 3

Input

nums = [4,5,6,7,0,1,2]
target = 0

The screenshot shows a LeetCode submission page for problem 4. The code is a Java implementation of the fourSum function. It uses a sorting step followed by a nested loop approach to find all quadruplets that sum up to the target value. The submission was accepted with 294/294 testcases passed, submitted by Saurav.raj01 at Feb 23, 2026 23:56. The runtime is 19 ms (Beats 80.59%) and memory usage is 45.83 MB (Beats 63.36%). A complexity analysis is available. The code editor shows the Java code, and the results section displays the accepted status, runtime (1 ms), and two test cases (Case 1 and Case 2) with their respective inputs and outputs.

```
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            for (int j = i + 1; j < n - 2; j++) {
12                for (int k = j + 1; k < n - 1; k++) {
13                    for (int l = k + 1; l < n; l++) {
14                        if (nums[i] + nums[j] + nums[k] + nums[l] == target) {
15                            List<Integer> quad = new ArrayList<>();
16                            quad.add(nums[i]);
17                            quad.add(nums[j]);
18                            quad.add(nums[k]);
19                            quad.add(nums[l]);
20                            result.add(quad);
21                        }
22                    }
23                }
24            }
25        }
26
27        return result;
28    }
29 }
```

Problem List < > <> Submit Ctrl Enter Code

Description Accepted Editorial Solutions Submissions

Saurav.raj01 submitted at Feb 23, 2026 23:55

Accepted 88 / 88 testcases passed

Runtime: 131 ms | Beats 71.40% Memory: 43.01 MB | Beats 61.48%

Analyze Complexity

Runtime distribution chart showing a sharp peak at 127ms.

Java code:

```
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        }
13    }
14
15    private boolean dfs(char[][] board, String word, int i, int j, int index) {
16        if (index == word.length()) {
17            return true;
18        }
19
20        if (i < 0 || i >= m || j < 0 || j >= n || board[i][j] != word.charAt(index)) {
21            return false;
22        }
23
24        char temp = board[i][j];
25        board[i][j] = '#';
26
27        boolean result = dfs(board, word, i + 1, j, index + 1) ||
28                      dfs(board, word, i - 1, j, index + 1) ||
29                      dfs(board, word, i, j + 1, index + 1) ||
30                      dfs(board, word, i, j - 1, index + 1);
31
32        board[i][j] = temp;
33
34        return result;
35    }
36}
```

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 =

```
"ABCED"
```

Array < > X

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted: 10 / 10 testcases passed

Saurav.raj01 submitted at Feb 23, 2026 23:52

Runtime: 1 ms | Beats 87.91% | Analyze Complexity

Memory: 44.12 MB | Beats 76.20% | Analyze Complexity

Code: Java

```
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
12             for (int i = 0; i < n; i++) if ((mask & (1 << i)) != 0)
13                 subset.add(nums[i]);
14
15             result.add(subset);
16         }
17     }
18 }
```

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 < > ⌂

Description Accepted Editorial Solutions Submissions

Accepted 89 / 89 testcases passed

Saurav.raj01 submitted at Feb 23, 2026 23:51

Editorial Solution

Runtime 0 ms Beats 100.00% Memory 43.63 MB Beats 42.16%

Analyze Complexity

100% 75% 50% 25% 0%

Runtime: 0ms, 1ms, 2ms, 3ms, 4ms

Memory: 0, 1ms, 2ms, 3ms, 4ms

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) {
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}
```

Java Auto

Code

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]

Problem List < > Submit Premium

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted 133 / 133 testcases passed
Saurav.raj01 submitted at Feb 23, 2026 23:49

Runtime: 0 ms | Beats 100.00%
Memory: 43.75 MB | Beats 91.32%

Analyze Complexity

150%
100%
50%
0%

1ms 2ms 3ms 4ms

Code Java

```
1 class Solution {
2     public boolean searchMatrix(int[][] matrix, int target) {
3         if (matrix == null || matrix.length == 0) return false;
4         int m = matrix.length;
5         int n = matrix[0].length;
```

Code

Java Auto

```
6         int left = 0;
7         int right = m * n - 1;
8
9         while (left <= right) {
10             int mid = left + (right - left) / 2;
11
12             int val = matrix[mid / n][mid % n];
13
14             if (val == target) return true;
15
16             if (val < target) left = mid + 1;
17             else right = mid - 1;
18         }
19
20         return false;
21 }
```

Saved

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 ⌂ ⌂ Premium

Description Accepted X Editorial Solutions Submissions

← All Submissions

Accepted 202 / 202 testcases passed
Saurav.raj01 submitted at Feb 23, 2026 23:47

Runtime: 1 ms | Beats 60.39% Memory: 47.71 MB | Beats 13.78%

Analyze Complexity

0% 25% 50% 75% 0% 1ms 2ms 3ms 4ms

Code Java

```
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         for (int i = 0; i < m; i++) {  
16             if (matrix[i][0] == 0) {  
17                 firstColZero = true;  
18                 break;  
19             }  
20         }  
21         for (int i = 1; i < m; i++) {  
22             for (int j = 1; j < n; j++) {  
23                 if (matrix[i][j] == 0) {  
24                     matrix[i][0] = 0;  
25                     matrix[0][j] = 0;  
26                 }  
27             }  
28         }  
29         for (int i = 1; i < m; i++) {  
30             for (int j = 1; j < n; j++) {  
31                 if (matrix[i][0] == 0 || matrix[0][j] == 0) {  
32                     matrix[i][j] = 0;  
33                 }  
34             }  
35         }  
36         if (firstRowZero) {  
37             for (int j = 0; j < n; j++) {  
38                 matrix[0][j] = 0;  
39             }  
40         }  
41         if (firstColZero) {  
42             for (int i = 0; i < m; i++) {  
43                 matrix[i][0] = 0;  
44             }  
45         }  
46     }  
47 }
```

Code 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]]
```

Accepted 114 / 114 testcases passed

Saurav_raj01 submitted at Feb 23, 2026 23:45

Runtime: 0 ms Beats 100.00% Memory: 43.52 MB Beats 38.32%

Analyze Complexity

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    }
}
```

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

```
digits =
[1,2,3]
```

Output

```
[1,2,4]
```

Problem List < > ✎

Description Accepted ✎ Editorial Solutions Submissions

All Submissions

Accepted 128 / 128 testcases passed

Saurav_raj01 submitted at Feb 23, 2026 23:44

Runtime 7 ms Beats 76.62% Memory 49.64 MB Beats 75.18%

Analyze Complexity

0% 20% 40%

0% 1ms 21ms 31ms 41ms

0% 1ms 21ms 31ms 41ms

Code Java Auto

```
16     map.put(key, new ArrayList<>());
17 }
18
19     map.get(key).add(word);
20 }
21
22 }
23
24 }
25
26 }
```

Saved Line 26, Col 1

Testcase 1 Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

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

Output

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

Code Java

```
1 import java.util.*;
2
3 class Solution {
4     public List<List<String>> groupAnagrams(String[] strs) {
5 }
```

Problem List < > ✎

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted 110 / 110 testcases passed.

Saurav_raj01 submitted at Feb 23, 2026 23:43

Runtime 1 ms Beats 99.60% Analyze Complexity

Memory 47.42 MB Beats 36.30%

0% 100% 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;
```

Code

```
12     end = far;
13 }
14 }
15 }
16 }
17 }
18 }
19 }
```

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

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

Output

2

Problem List < > Submit Premium

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted 176 / 176 testcases passed
Saurav.raj01 submitted at Feb 23, 2026 23:42

Runtime: 6 ms | Beats 73.38%
Memory: 47.46 MB | Beats 11.25%
[Analyze Complexity](#)

0% 20% 40%

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<>();
```

Code

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);

Saved

Testcase | Test Result

Accepted Runtime: 1 ms

Case 1 Case 2

Input

```
candidates =
[10,1,2,7,6,1,5]
```

target = 8

Problem List < > ✎

Submit

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted 160 / 160 testcases passed.

Saurav_raj01 submitted at Feb 23, 2026 23:38

Runtime 2 ms | Beats 95.12% Memory 46.04 MB | Beats 19.62%

Analyze Complexity

75%
50%
25%
0% 1ms 2ms 3ms 4ms 5ms

Code Java

```
for (int i = start; i < candidates.length; i++) {  
    temp.add(candidates[i]);  
    backtrack(candidates, target - candidates[i], i, temp, result);  
    temp.remove(temp.size() - 1);  
}  
}
```

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

candidates = [2, 3, 6, 7]
target = 7

Array < > ✎

Description Accepted Editorial Solutions Submissions

Accepted 66 / 66 testcases passed

Saurav.raj01 submitted at Feb 23, 2026 23:37

Runtime: 0 ms | Beats 100.00% Memory: 44.96 MB | beats 27.16%

Analyze Complexity

Code

```
Java ▾ Auto
16     right = mid - 1;
17 }
18 }
19 }
20 }
21 }
22 }
23 }
```

Saved

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     }
}
```