

What's New

Find the Index of the First Occu

Find the Index of the First Occu

+

https://leetcode.com/problems/find-the-index-of-the-first-occurrence-in-a-string/description/

Problem List

Run

Submit

0

Premium

Description

Editorial

Solutions

Submissions

## 28. Find the Index of the First Occurrence in a String Solved

Easy

Topics

Companies

Given two strings `needle` and `haystack`, return the index of the first occurrence of `needle` in `haystack`, or `-1` if `needle` is not part of `haystack`.

**Example 1:**

**Input:** `haystack = "sadbutsad", needle = "sad"`  
**Output:** `0`  
**Explanation:** "sad" occurs at index 0 and 6. The first occurrence is at index 0, so we return 0.

**Example 2:**

**Input:** `haystack = "leetcode", needle = "leeto"`  
**Output:** `-1`  
**Explanation:** "leeto" did not occur in "leetcode". so we return -1.

6.4K

318

27 Online

Code

C++

Auto

```
1 #include <string>
2 using namespace std;
3
4 class Solution {
5 public:
6     int strStr(string haystack, string needle) {
7         // Find the position of the first occurrence of needle in haystack
8         size_t pos = haystack.find(needle);
9
10        // If found, return the position; otherwise, return -1
11        return (pos != string::npos) ? pos : -1;
12    }
13 };
14
```

Saved

Ln 1, Col 1

Testcase

Test Result

1

Search

ENG IN

08:45

20-01-2025

What's New

Bitwise AND of Numbers Range

Find the Index of the First Occu

+

https://leetcode.com/problems/bitwise-and-of-numbers-range/

🔍 ☆ 👤 ⋮

Problem List < > 🔍

Run Submit ⌚ 📄

🔧 ⚙️ 🔥 0 👤 Premium

Description Editorial Solutions Submissions

< >

</> Code

201. Bitwise AND of Numbers Range

Medium Topics Companies

Given two integers `left` and `right` that represent the range `[left, right]`, return the bitwise AND of all numbers in this range, inclusive.

**Example 1:**

Input: `left = 5, right = 7`  
Output: `4`

**Example 2:**

Input: `left = 0, right = 0`  
Output: `0`

**Example 3:**

Input: `left = 1, right = 2147483647`  
Output: `0`

4.1K 81 ☆ 📄 ?

3 Online

C++ Auto

1 class Solution {  
2 public:  
3 int rangeBitwiseAnd(int left, int right) {  
4 int shift=0;  
5 while(left<right){  
6 left>>=1;  
7 right>>=1;  
8 }  
9 return left;  
10 }

Saved Ln 10, Col 28

Testcase Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

left =  
5

right =

What's New

Sqrt(x) - LeetCode

Find the Index of the First Occu

+

https://leetcode.com/problems/sqrtx/

Problem List

Run

Submit

88

0

Premium

Description

Editorial

Solutions

Submissions

69. Sqrt(x)

Easy

Topics

Companies

Hint

Given a non-negative integer  $x$ , return the square root of  $x$  rounded down to the nearest integer. The returned integer should be **non-negative** as well.

You **must not use** any built-in exponent function or operator.

- For example, do not use `pow(x, 0.5)` in c++ or `x ** 0.5` in python.

Example 1:

Input:  $x = 4$

Output: 2

Explanation: The square root of 4 is 2, so we return 2.

Example 2:

Input:  $x = 8$

Output: 2

Explanation: The square root of 8 is 2.82842..., and since we round it down to the nearest integer, 2 is returned.

8.6K

273

16 Online

Code

C++

Auto

```
1 #include <iostream>
2 using namespace std;
3
4 class Solution {
5 public:
6     int mySqrt(int x) {
7         if (x == 0 || x == 1) return x; // Handle edge cases
8
9         int left = 0, right = x, result = 0;
10
11         while (left <= right) {
12             long long mid = left + (right - left) / 2; // Use long long to avoid
overflow
13             long long square = mid * mid;
14
15             if (square == x) {
16                 return mid; // Perfect square
17             } else if (square < x) {
```

Saved

Ln 28, Col 1

Testcase

Test Result

Accepted

Runtime: 0 ms

Case 1

Case 2



What's New

Largest Number - LeetCode

Find the Index of the First Occu

+

https://leetcode.com/problems/largest-number/

67 🔍 ☆ 👤 ⋮

Problem List < > ✕

Run Submit ⌚ 📄

88 ⚙️ 💧 0 👤 Premium

Description Editorial Solutions Submissions

179. Largest Number

Medium Topics Companies

Given a list of non-negative integers `nums`, arrange them such that they form the largest number and return it.

Since the result may be very large, so you need to return a string instead of an integer.

**Example 1:**

**Input:** `nums = [10,2]`  
**Output:** `"210"`

**Example 2:**

**Input:** `nums = [3,30,34,5,9]`  
**Output:** `"9534330"`

**Constraints:**

- `1 <= nums.length <= 100`
- `0 <= nums[i] <= 109`

9K 186 ☆ 📄 ⓘ

4 Online

Code

C++ Auto

```
1 #include <vector>
2 #include <string>
3 #include <algorithm>
4 using namespace std;
5
6 class Solution {
7 public:
8     string largestNumber(vector<int>& nums) {
9         vector<string> strNums;
10        for (int num : nums) {
11            strNums.push_back(to_string(num));
12        }
13
14        auto compare = [](const string& a, const string& b) {
15            return a + b > b + a;
16        };
17
18        sort(strNums.begin(), strNums.end(), compare);
19
20        string result;
21        for (const string& str : strNums) {
22            result += str;
23        }
24    }
25};
```

Saved Upgrade to Cloud Saving Ln 32, Col 1

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

## 20. Valid Parentheses

Easy Topics Companies Hint

Given a string `s` containing just the characters `'('`, `')'`, `'{'`, `'}'`, `'['` and `']'`, determine if the input string is valid.

An input string is valid if:

- Open brackets must be closed by the same type of brackets.
- Open brackets must be closed in the correct order.
- Every close bracket has a corresponding open bracket of the same type.

Example 1:

Input: `s = "()"`

Output: `true`

Example 2:

Input: `s = "()[]{}"`

Output: `true`

Example 3:

25.1K 457

68 Online

Code

C++ Auto

```
1 #include <iostream>
2 #include <stack>
3 #include <string>
4 using namespace std;
5
6 class Solution {
7 public:
8     bool isValid(string s) {
9         stack<char> stack;
10
11         for (char c : s) {
12             // If the character is an opening bracket, push it onto the stack
13             if (c == '(' || c == '{' || c == '[') {
14                 stack.push(c);
15             }
16             // If the character is a closing bracket, check if it matches the top of the stack
17             else if (c == ')' || c == '}' || c == ']') {
18                 if (stack.empty()) {
19                     return false; // No opening bracket to match with
20                 }
21                 char top = stack.top();
22                 if ((c == ')' && top == '(') || (c == '}' && top == '{') || (c == ']' && top == '[')) {
23                     stack.pop();
24                 } else {
25                     return false;
26                 }
27             }
28         }
29         return stack.empty();
30     }
31 }
```

Saved

Ln 35, Col 1

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3 Case 4

Input



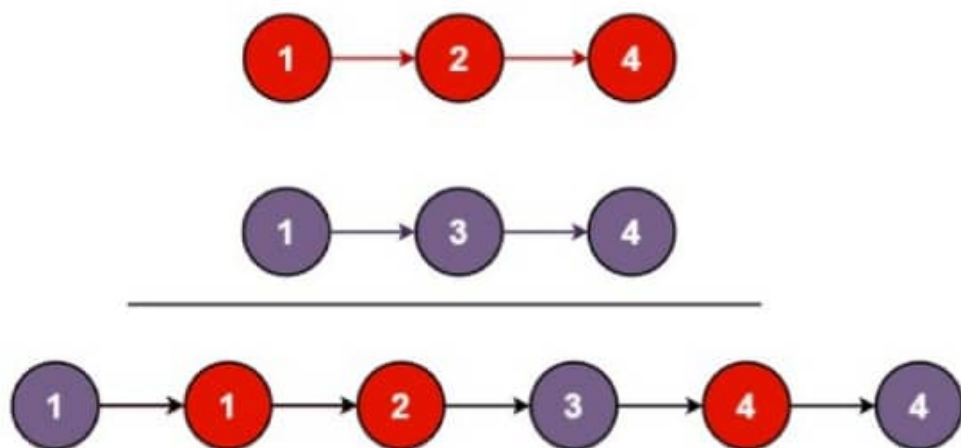
## 21. Merge Two Sorted Lists

You are given the heads of two sorted linked lists `list1` and `list2`.

Merge the two lists into one **sorted** list. The list should be made by splicing together the nodes of the first two lists.

Return the head of the merged linked list.

**Example 1:**



22.7K 393

● 63 Online

Code

C++   Auto

```

1 #include <iostream>
2 #include <vector>
3 // Include the ListNode header
4 using namespace std;
5
6 class Solution {
7 public:
8     ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {
9         ListNode* dummy = new ListNode(0); // Create a dummy node to start the merged list
10        ListNode* current = dummy; // Pointer to build the new list
11
12        // Traverse both lists
13        while (list1 != nullptr && list2 != nullptr) {
14            if (list1->val <= list2->val) {
15                current->next = list1; // Attach list1 node to merged list
16                list1 = list1->next;    // Move the list1 pointer forward
17            } else {
18                current->next = list2; // Attach list2 node to merged list
19                list2 = list2->next;    // Move the list2 pointer forward
20            }
21            current = current->next; // Move the current pointer forward in the merged list

```

Ln 59, Col 1

Testcase > Test Result

Accepted Runtime: 0 ms

- Case 1
- Case 2
- Case 3

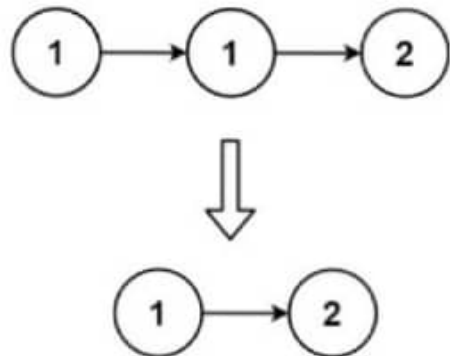
input

## 83. Remove Duplicates from Sorted List

Easy Topics Companies

Given the `head` of a sorted linked list, delete all duplicates such that each element appears only once. Return the linked list *sorted* as well.

Example 1:



Input: `head = [1,1,2]`

Output: `[1,2]`

Example 2:



9K 103 1 1

11 Online

Code

C++ Auto

```
1 #include <iostream>
2 using namespace std;
3
4 // Assume ListNode structure is already defined and precompiled.
5
6 class Solution {
7 public:
8     ListNode* deleteDuplicates(ListNode* head) {
9         ListNode* current = head;
10        while (current != nullptr && current->next != nullptr) {
11            if (current->val == current->next->val) {
12                // Skip the duplicate node
13                current->next = current->next->next;
14            } else {
15                // Move to the next node
16                current = current->next;
17            }
18        }
19        return head;
20    }
21 };
22
```

Saved

Ln 34, Col 1

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input



Search



Scanned with OKEN Scanner

## 162. Find Peak Element

Medium Topics Companies

A peak element is an element that is strictly greater than its neighbors.

Given a 0-indexed integer array `nums`, find a peak element, and return its index. If the array contains multiple peaks, return the index to **any of the peaks**.

You may imagine that `nums[-1] = nums[n] = -∞`. In other words, an element is always considered to be strictly greater than a neighbor that is outside the array.

You must write an algorithm that runs in  $O(\log n)$  time.

### Example 1:

**Input:** `nums = [1,2,3,1]`

**Output:** 2

**Explanation:** 3 is a peak element and your function should return the index number 2.

### Example 2:

**Input:** `nums = [1,2,1,3,5,6,4]`

**Output:** 5

**Explanation:** Your function can return either index number 1 where the peak element is 2, or index number 5 where the peak element is 6.

12.8K 293

25 Online

### Code

C++ Auto

```
using namespace std;

class Solution {
public:
    int findPeakElement(vector<int>& nums) {
        int left = 0, right = nums.size() - 1;

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

            if (nums[mid] > nums[mid + 1]) {
                right = mid;
            } else {
                left = mid + 1;
            }
        }

        return left;
    }
};
```

Saved

Ln 23, Col 1

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input



Search



ENG

IN



08:53

20-01-2025



Scanned with OKEN Scanner



## 94. Binary Tree Inorder Traversal

Easy Topics Companies

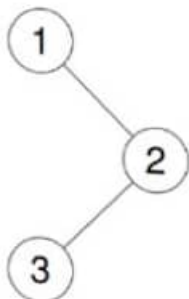
Given the `root` of a binary tree, return the *inorder traversal* of its nodes' values.

### Example 1:

Input: `root = [1,null,2,3]`

Output: `[1,3,2]`

Explanation:



13.9K 187

21 Online

</> Code

C++ Auto

```
1 #include <iostream>
2 #include <vector>
3
4
5 using namespace std;
6
7 class Solution {
8 public:
9     void inorderTraversalHelper(TreeNode* root, vector<int>& result) {
10         if (root == nullptr) {
11             return;
12         }
13
14         // Traverse left subtree
15         inorderTraversalHelper(root->left, result);
16
17         // Visit the root node
18         result.push_back(root->val);
19
20         // Traverse right subtree
21         inorderTraversalHelper(root->right, result);
22     }
23
24     vector<int> inorderTraversal(TreeNode* root) {
25         vector<int> result;
```

Saved

Testcase Test Result

Accepted Runtime: 0 ms

Ln 31, Col 1

## 51. N-Queens

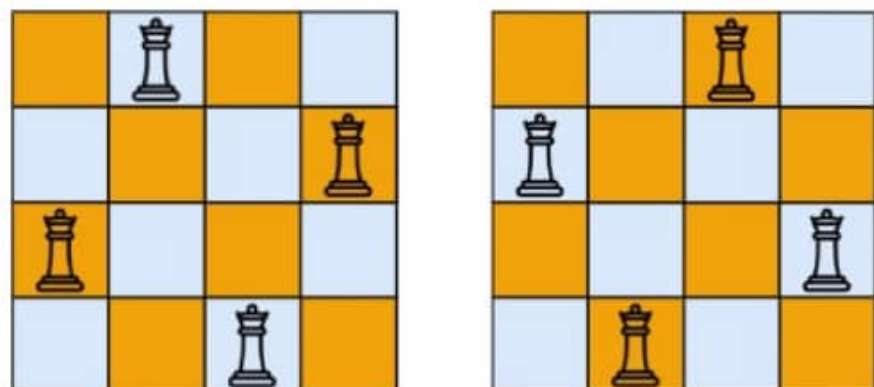
Hard Topics Companies

The **n-queens** puzzle is the problem of placing  $n$  queens on an  $n \times n$  chessboard such that no two queens attack each other.

Given an integer  $n$ , return all distinct solutions to the **n-queens puzzle**. You may return the answer in any order.

Each solution contains a distinct board configuration of the  $n$ -queens' placement, where `'Q'` and `'.'` both indicate a queen and an empty space, respectively.

Example 1:



12.9K 116

15 Online

Code

C++ Auto

```
1 #include <vector>
2 #include <string>
3 using namespace std;
4
5 class Solution {
6 public:
7     vector<vector<string>> solveNQueens(int n) {
8         vector<vector<string>> solutions;
9         vector<string> board(n, string(n, '.')); // Initialize an empty n x n board
10        vector<int> leftRow(n, 0), upperDiag(2 * n - 1, 0), lowerDiag(2 * n - 1, 0);
11        backtrack(0, n, board, solutions, leftRow, upperDiag, lowerDiag);
12        return solutions;
13    }
14
15 private:
16     void backtrack(int col, int n, vector<string>& board, vector<vector<string>>& solutions,
17                    vector<int>& leftRow, vector<int>& upperDiag, vector<int>& lowerDiag) {
18         if (col == n) {
19             solutions.push_back(board);
20             return;
21         }
22
23         for (int row = 0; row < n; ++row) {
24             if (leftRow[row] == 0 && upperDiag[row + col] == 0 && lowerDiag[row - col + n - 1] ==
25                 0) {
26                 board[row][col] = 'Q';
27                 leftRow[row] = upperDiag[row + col] = lowerDiag[row - col + n - 1] = 1;
28                 backtrack(col + 1, n, board, solutions, leftRow, upperDiag, lowerDiag);
29                 board[row][col] = '.';
30                 leftRow[row] = upperDiag[row + col] = lowerDiag[row - col + n - 1] = 0;
31             }
32         }
33     }
34 }
```

Saved

Ln 40, Col 1

Testcase Test Result



Search



Scanned with OKEN Scanner