

CSA0672 - Google Sheets x Two Sum - LeetCode x +

leetcode.com/problems/two-sum/description/

Problem List < > Run Submit

Description Editorial Solutions Accepted x Submissions

1. Two Sum

Easy Topics Companies Hint

Given an array of integers `nums` and an integer `target`, return *indices* of the two numbers such that they add up to `target`.

You may assume that each input would have **exactly one solution**, and you may not use the same element twice.

You can return the answer in any order.

Example 1:

Input: `nums = [2,7,11,15]`, `target = 9`
Output: `[0,1]`
Explanation: Because `nums[0] + nums[1] == 9`, we return `[0, 1]`.

Example 2:

Input: `nums = [3,2,4]`, `target = 6`
Output: `[1,2]`

Example 3:

Input: `nums = [3,3]`, `target = 6`
Output: `[0,1]`

56.5K 850

Code

```
C++ v Auto
1 class Solution {
2 public:
3     vector<int> twoSum(vector<int>& nums, int target) {
4
5         for(int i=0;i<nums.size();i++){
6             int sum=0;
7             for(int j=i+1;j<nums.size();j++){
8                 sum=nums[i]+nums[j];
9                 if(sum==target){
10                     return {i,j};
11                     break;
12                 }
13             }
14         }
15
16         return {};
17     }
18
19     return {};
20 }
21
22 }
```

Saved Ln 22, Col 3

Testcase Test Result

23°C Cloudy Search

CSA0672 - Google Sheets x Add Two Numbers - LeetCode x +

leetcode.com/problems/add-two-numbers/

Problem List < > Run Submit

Description Editorial Solutions Submissions

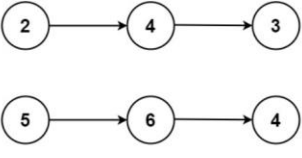
2. Add Two Numbers

Medium Topics Companies

You are given two **non-empty** linked lists representing two non-negative integers. The digits are stored in **reverse order**, and each of their nodes contains a single digit. Add the two numbers and return the sum as a linked list.

You may assume the two numbers do not contain any leading zero, except the number 0 itself.

Example 1:



30.8K 578

Code

```
C++ v Auto
1 class Solution {
2 public:
3     ListNode* addTwoNumbers(ListNode* l1, ListNode* l2) {
4         ListNode* dummy=new ListNode();
5         ListNode* temp=dummy;
6         int carry=0;
7         while(l1!=NULL || l2!=NULL || carry){
8             int sum=0;
9             if(l1!=NULL){
10                 sum+=l1->val;
11                 l1=l1->next;
12             }
13             if(l2!=NULL){
14                 sum+=l2->val;
15                 l2=l2->next;
16             }
17             sum+=carry;
18             carry=sum/10;
19             ListNode* newnode=new ListNode(sum%10);
20             temp->next=newnode;
21             temp=temp->next;
22         }
23         return dummy->next;
24     }
25 };
26 }
```

Saved Ln 1, Col 1

Testcase Test Result

https://leetcode.com/problems/longest-substring-without-repeating-characters

23°C Cloudy Search

CSA0672 - Google Sheets x Longest Substring Without Rep: x +

leetcode.com/problems/longest-substring-without-repeating-characters/description/

Problem List < > > Run Submit

Description Editorial Solutions Accepted x Submissions

3. Longest Substring Without Repeating Characters

Solved

Medium Topics Companies Hint

Given a string `s`, find the length of the **longest substring** without repeating characters.

Example 1:

Input: `s = "abcabcbb"`
Output: 3
Explanation: The answer is "abc", with the length of 3.

Example 2:

Input: `s = "bbbbb"`
Output: 1
Explanation: The answer is "b", with the length of 1.

Example 3:

Input: `s = "pwwkew"`
Output: 3
Explanation: The answer is "wke", with the length of 3. Notice that the answer must be a substring, "pwke" is a subsequence and not a substring.

39.4K 331

```
1 class Solution {
2 public:
3     int lengthOfLongestSubstring(string s) {
4         unordered_map<char,int> m;
5         int left=0;
6         int right=0;
7         int maxSize=0;
8         for(auto x:m) cout<<x.first<<" "<<x.second<<endl;
9         while(right<s.size()){
10             m[s[right]]++;
11             while(m[s[right]]>1){
12                 m[s[left]]--;
13                 left++;
14             }
15             maxSize=max(maxSize,right-left+1);
16             right++;
17         }
18         return maxSize;
19     }
20 };
```

Saved Ln 1, Col 1

Testcase Test Result

23°C Light rain Search

CSA0672 - Google Sheets x Median of Two Sorted Arrays - x +

leetcode.com/problems/median-of-two-sorted-arrays/description/

Problem List < > > Run Submit

Description Editorial Solutions Accepted x Submissions

4. Median of Two Sorted Arrays

Solved

Hard Topics Companies

Given two sorted arrays `nums1` and `nums2` of size `m` and `n` respectively, return the **median** of the two sorted arrays.

The overall run time complexity should be $O(\log(m+n))$.

Example 1:

Input: `nums1 = [1,3], nums2 = [2]`
Output: 2.00000
Explanation: merged array = [1,2,3] and median is 2.

Example 2:

Input: `nums1 = [1,2], nums2 = [3,4]`
Output: 2.50000
Explanation: merged array = [1,2,3,4] and median is (2 + 3) / 2 = 2.5.

Constraints:

- `nums1.length == m`

28.1K 418

```
1 class Solution {
2 public:
3     double findMedianSortedArrays(vector<int>& arr1, vector<int>& arr2) {
4         int n=arr1.size()+arr2.size();
5         vector<int> arr3(n);
6         merge(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), arr3.begin());
7         int al=arr3.size();
8         sort(arr3.begin(),arr3.end());
9         if(al%2==1){
10             int p=((al)/2);
11             return arr3[p];
12         }
13         else{
14             int o=al/2;
15             double k=(arr3[o]+arr3[o-1]);
16             return ((k/2));
17         }
18     }
19 };
20
```

Saved Ln 7, Col 28

Testcase Test Result

Finance headline Foreign portfolio...

Search

ENG IN 09:50 PM 05-06-2024

CSA0672 - Google Sheets x Longest Palindromic Substring x +

leetcode.com/problems/longest-palindromic-substring/description/

5. Longest Palindromic Substring

Medium Topics Companies Hint

Given a string *s*, return the longest *palindromic substring* in *s*.

Example 1:
Input: *s* = "babad"
Output: "bab"
Explanation: "aba" is also a valid answer.

Example 2:
Input: *s* = "cbbd"
Output: "bb"

Constraints:

- $1 \leq s.length \leq 1000$
- s* consist of only digits and English letters.

```
1 class Solution {
2 public:
3     string longestPalindrome(string s) {
4         string res = "";
5         int bestLeftRight[2] = {0};
6         int left = 0;
7         int right = 0;
8         for(int center = 1; center < s.size() * 2 + 1; center++){
9
10            if(center % 2){
11                left = (center/2) - 1;
12                right = (center/2) + 1;
13            }else{
14                left = ((center-1)/2);
15                right = (center/2);
16            }
17
18            while(left >= 0 && right < s.size() && s[left] == s[right]){
19                left--;
20                right++;
21            }
22
23            if((right - left - 2) >= (bestLeftRight[1] - bestLeftRight[0])){
24                bestLeftRight[0] = left+1;
25                bestLeftRight[1] = right-1;
26            }
```

Saved Ln 13, Col 42

Testcase Test Result

IND - IRE Live

CSA0672 - Google Sheets x Zigzag Conversion - LeetCode x +

leetcode.com/problems/zigzag-conversion/description/

6. Zigzag Conversion

Medium Topics Companies

The string "PAYPALISHIRING" is written in a zigzag pattern on a given number of rows like this: (you may want to display this pattern in a fixed font for better legibility)

```
P A H N
A P L S I I G
Y I R
```

And then read line by line: "PAHNAPLSIIGYIR"

Write the code that will take a string and make this conversion given a number of rows:

```
string convert(string s, int numRows);
```

Example 1:
Input: *s* = "PAYPALISHIRING", *numRows* = 3
Output: "PAHNAPLSIIGYIR"

Example 2:
Input: *s* = "PAYPALISHIRING", *numRows* = 4
Output: "PINALSIGYAHRPI"

```
1 class Solution {
2 public:
3     string convert(string s, int numRows) {
4         string str[numRows], ans;
5         int j = 0;
6         bool x = false;
7
8         if(numRows >= s.size() || numRows == 1)
9             return s;
10
11         for(int i = 0; i < s.size(); i++){
12             str[j] += s[i];
13             if(j == 0)
14                 x = false;
15             if(j == numRows - 1)
16                 x = true;
17
18             if(x == true)
19                 j--;
20             else
21                 j++;
22         }
23         for(int i = 0; i < numRows; i++)
24             ans += str[i];
25
26         return ans;
27     }
```

Saved Ln 1, Col 1

Testcase Test Result

CSA0672 - Google Sheets x Reverse Integer - LeetCode x +

leetcode.com/problems/reverse-integer/description/

Problem List < > 🔍 Run Submit 📄

Description Accepted x Editorial Solutions Submissions

7. Reverse Integer

Medium Topics Companies

Solved

Given a signed 32-bit integer x , return x with its digits reversed. If reversing x causes the value to go outside the signed 32-bit integer range $[-2^{31}, 2^{31} - 1]$, then return 0.

Assume the environment does not allow you to store 64-bit integers (signed or unsigned).

Example 1:
Input: $x = 123$
Output: 321

Example 2:
Input: $x = -123$
Output: -321

Example 3:
Input: $x = 120$
Output: 21

```
1 class Solution {
2 public:
3     int reverse(int x) {
4         long rev=0;
5         while(x){
6             rev*=10+x%10;
7             x=x/10;
8         }
9         if(rev>INT_MAX || rev<INT_MIN) return 0; return rev;
10    }
11 }
12
```

Saved Ln 10, Col 25

Testcase Test Result

23°C Cloudy Search

CSA0672 - Google Sheets x String to Integer (atoi) - LeetCode x +

leetcode.com/problems/string-to-integer-atoi/description/

Problem List < > 🔍 Run Submit 📄

Description Accepted x Editorial Solutions Submissions

8. String to Integer (atoi)

Medium Topics Companies

Solved

Implement the `myAtoi(string s)` function, which converts a string to a 32-bit signed integer.

The algorithm for `myAtoi(string s)` is as follows:

- Whitespace:** Ignore any leading whitespace (" ").
- Signedness:** Determine the sign by checking if the next character is '-' or '+', assuming positivity is neither present.
- Conversion:** Read the integer by skipping leading zeros until a non-digit character is encountered or the end of the string is reached. If no digits were read, then the result is 0.
- Rounding:** If the integer is out of the 32-bit signed integer range $[-2^{31}, 2^{31} - 1]$, then round the integer to remain in the range. Specifically, integers less than -2^{31} should be rounded to -2^{31} , and integers greater than $2^{31} - 1$ should be rounded to $2^{31} - 1$.

Return the integer as the final result.

Example 1:
Input: $s = "42"$
Output: 42

```
1 class Solution {
2 public:
3     int myAtoi(string s)
4     {
5         int i=0;
6         int sign=1;
7         long ans=0;
8         while(i<s.length() && s[i]!=' ')
9             i++;
10        if(s[i]=='-')
11        {
12            sign=-1;
13            i++;
14        }
15        else if(s[i]=='+')
16        {
17            i++;
18        }
19        while(i<s.length())
20        {
21            if(s[i]>='0' && s[i]<='9')
22            {
23                ans=ans*10+(s[i]-'0');
24                if(ans>INT_MAX && sign==1)
25                    return INT_MAX;
26                else if(ans>INT_MAX && sign==1)
27                    return INT_MAX;
28            }
29            i++;
30        }
31        return ans*sign;
32    }
33 }
```

Saved Ln 33, Col 3

Testcase Test Result

CSA0672 - Google Sheets x Palindrome Number - LeetCode x +

leetcode.com/problems/palindrome-number/description/

Problem List < > > Run Submit

Description Editorial Solutions Accepted x Submissions

9. Palindrome Number

Easy Topics Companies Hint

Given an integer x , return `true` if x is a **palindrome** and `false` otherwise.

Example 1:
Input: $x = 121$
Output: `true`
Explanation: 121 reads as 121 from left to right and from right to left.

Example 2:
Input: $x = -121$
Output: `false`
Explanation: From left to right, it reads -121. From right to left, it becomes 121-. Therefore it is not a palindrome.

Example 3:
Input: $x = 10$
Output: `false`
Explanation: Reads 01 from right to left. Therefore it is not a palindrome.

12.5K 371 ☆

`</> Code`

```
C++ v Auto
1 class Solution {
2 public:
3     bool isPalindrome(int x) {
4         if (x < 0) {
5             return false;
6         }
7
8         long long reversed = 0;
9         long long temp = x;
10
11         while (temp != 0) {
12             int digit = temp % 10;
13             reversed = reversed * 10 + digit;
14             temp /= 10;
15         }
16
17         return (reversed == x);
18     }
19 };
```

Saved Ln 1, Col 1

Testcase Test Result

23°C Cloudy Search

CSA0672 - Google Sheets x Regular Expression Matching - x +

leetcode.com/problems/regular-expression-matching/description/

Problem List < > > Run Submit

Description Editorial Solutions Accepted x Submissions

10. Regular Expression Matching

Hard Topics Companies

Given an input string s and a pattern p , implement regular expression matching with support for `'.'` and `'*'` where:

- `'.'` Matches any single character.
- `'*'` Matches zero or more of the preceding element.

The matching should cover the **entire** input string (not partial).

Example 1:
Input: $s = "aa"$, $p = "a"$
Output: `false`
Explanation: "a" does not match the entire string "aa".

Example 2:
Input: $s = "aa"$, $p = "a*"$
Output: `true`
Explanation: `'*'` means zero or more of the preceding element, `'a'`. Therefore, by repeating `'a'` once, it becomes `"aa"`.

12.1K 308 ☆

`</> Code`

```
C++ v Auto
1 class Solution {
2 public:
3     bool isMatch(string s, string p) {
4         int m = s.size(), n = p.size();
5         vector<vector<bool>>> dp(m + 1, vector<bool>(n + 1, false));
6         dp[0][0] = true;
7         for (int i = 0; i <= m; i++) {
8             for (int j = 1; j <= n; j++) {
9                 if (p[j - 1] == '*') {
10                     dp[i][j] = dp[i][j - 2] || (i && dp[i - 1][j] && (s[i - 1] == p[j - 2] || p[j - 2] == '.'));
11                 } else {
12                     dp[i][j] = i && dp[i - 1][j - 1] && (s[i - 1] == p[j - 1] || p[j - 1] == '.');
13                 }
14             }
15         }
16         return dp[m][n];
17     }
18 };
```

Saved Ln 18, Col 3

Testcase Test Result

23°C Cloudy Search