

Q 1(Easy): Two Sum

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

```
class Solution {  
    public int[] twoSum(int[] nums, int target) {  
  
        for(int i=0;i<nums.length;i++)  
        {  
            // System.out.println("value of i:" +i);  
            for(int j=i+1;j<nums.length;j++)  
            {  
                // System.out.println("value of J:" +j);  
                if(nums[j] == target-nums[i])  
                {  
                    return new int[]{i,j};  
                }  
            }  
        }  
        return new int[] {};  
    }  
}
```

The screenshot shows the LeetCode 'Two Sum' problem page. The solution is written in Java and uses a nested loop to find two numbers that add up to the target. The solution is marked as 'Accepted' with a runtime of 0 ms. The page also shows a list of 'Next challenges' including '3Sum', '4Sum', 'Two Sum II - Input Array Is Sorted', 'Two Sum III - Data structure design', 'Subarray Sum Equals K', 'Two Sum IV - Input is a BST', 'Two Sum Less Than K', 'Max Number of K-Sum Pairs', 'Count Good Meals', 'Count Number of Pairs With Absolute Difference K', and 'Number of Pairs of Strings With Concatenation Equal to Target'.

Q 2(Medium): Add Two Numbers

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.

```
/**
```

```
* Definition for singly-linked list.
```

```
* public class ListNode {
```

```
*     int val;
```

```
*     ListNode next;
```

```
*     ListNode() {}
```

```
*     ListNode(int val) { this.val = val; }
```

```
*   ListNode(int val, ListNode next) { this.val = val; this.next = next; }  
* }  
*/
```

```
class Solution {  
    public ListNode addTwoNumbers(ListNode l1, ListNode l2) {
```

```
        ListNode head = null;
```

```
        ListNode temp = null;
```

```
        int c = 0;
```

```
        while (l1 != null || l2 != null)
```

```
        {
```

```
            int sum = c;
```

```
            if (l1 != null)
```

```
            {
```

```
                sum += l1.val;
```

```
                l1 = l1.next;
```

```
            }
```

```
            if (l2 != null)
```

```
            {
```

```
                sum += l2.val;
```

```
                l2 = l2.next;
```

```
            }
```

```
            ListNode node = new ListNode(sum % 10);
```

```
            c = sum / 10;
```

```
            if (temp == null)
```

```
            {
```

```
                temp = head = node;
```

```

    }
    else
    {
        temp.next = node;
        temp = temp.next;
    }
}
if (c>0)
{
    temp.next = new ListNode(c);
}
return head;
}
}

```

Success Details >

Runtime: 3 ms, faster than 65.71% of Java online submissions for Add Two Numbers.

Memory Usage: 47.2 MB, less than 69.49% of Java online submissions for Add Two Numbers.

Next challenges:

- Multiply Strings
- Add Binary
- Sum of Two Integers
- Add Strings
- Add Two Numbers II
- Add to Array-Form of Integer
- Add Two Polynomials Represented as Linked Lists

Show off your acceptance: [f](#) [t](#) [in](#)

Time Submitted	Status	Runtime	Memory	Language

```

1  /**
2   * Definition for singly-linked list.
3   * public class ListNode {
4   *     int val;
5   *     ListNode next;
6   *     ListNode() {}
7   *     ListNode(int val) { this.val = val; }
8   *     ListNode(int val, ListNode next) { this.val = val; this.next = next; }
9   * }

```

Your previous code was restored from your local storage. [Reset to default](#)

Testcase Run Code Result Debuqger

Accepted Runtime: 0 ms

Your input [2,4,3]
[5,6,4]

Output [7,0,8] [Diff](#)

Expected [7,0,8]

Console Use Example Testcases

Run Code ^ Submit

Problems Pick One < Prev 2/2296 Next >

Type here to search

16:53 08-06-2022

Q 3(Hard): Given two sorted arrays `nums1` and `nums2` of size `m` and `n` respectively, return **the median** of the two sorted arrays.

```
class Solution {  
    public double findMedianSortedArrays(int[] nums1, int[] nums2) {  
  
        ArrayList<Integer> al = new ArrayList<>();  
        for(int i: nums1)  
        {  
            al.add(i);  
        }  
        for(int i: nums2)  
        {  
            al.add(i);  
        }  
        Collections.sort(al);  
        int n = al.size();  
        if(al.size()%2!=0)  
        {  
            return (double)al.get(n/2);  
        }  
        int p = al.get((n/2)-1);  
        System.out.println(p);  
  
        int q = al.get((n/2));  
        System.out.println(q);  
  
        System.out.println((double)(p+q)/2);  
        return (double)(p+q)/2;  
    }  
}
```

Median of Two Sorted Arrays - L...Upload files - rohit7kumar/LeetC...+

←→↻leetcode.com/problems/median-of-two-sorted-arrays/submissions/

New folderGmailYouTubeMaps1920x1080px | free...What are Selenium...Charging Stations F...Imported From IE

LeetCodeExploreProblemsInterviewContestDiscussStorePremium0

DescriptionSolutionDiscuss (999+)Submissions

SuccessDetails

Runtime: 30 ms, faster than 7.26% of Java online submissions for Median of Two Sorted Arrays.

Memory Usage: 51.1 MB, less than 7.87% of Java online submissions for Median of Two Sorted Arrays.

Next challenges:

Merge IntervalsDesign Circular Queue

Kth Missing Positive Number

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
----------------	--------	---------	--------	----------

JavaAutocomplete

```
21
22     int q = a1.get((n/2));
23     System.out.println(q);
24
25     System.out.println((double)(p+q)/2);
26     return (double)(p+q)/2;
27 }
28
29
```

Your previous code was restored from your local storage. [Reset to default](#)

TestcaseRun Code ResultDebugger

AcceptedRuntime: 0 ms

Your input

[1,3]

[2]

Output

2.00000

Diff

Expected

2.00000

ProblemsPick One4/2296Next>

ConsoleUse Example TestcasesRun Code ^Submit

Type here to search17:3208-06-202218