Description | Solution | Discuss (999+) | Submissions

349. Intersection of Two Arrays

Easy 2259 1810 Add to List Share

Given two integer arrays `nums1` and `nums2`, return *an array of their intersection*. Each element in the result must be **unique** and you may return the result in **any order**.

Example 1:

```
Input: nums1 = [1,2,2,1], nums2 = [2,2]
Output: [2]
```

Example 2:

```
Input: nums1 = [4,9,5], nums2 = [9,4,9,8,4]
Output: [9,4]
Explanation: [4,9] is also accepted.
```

Constraints:

- `1 <= nums1.length, nums2.length <= 1000`
- `0 <= nums1[i], nums2[i] <= 1000`

Accepted 589,606 | Submissions 867,649

Python3 | Autocomplete

```python
class Solution:
    def intersection(self, nums1: List[int], nums2: List[int]) -> List[int]:

        return set(nums1).intersection(set(nums2))

```

Your previous code was restored from your local storage. Reset to default

Description | Solution | Discuss (999+) | Submissions

1464. Maximum Product of Two Elements in an Array

Easy 746 131 Add to List Share

Given the array of integers `nums`, you will choose two different indices `i` and `j` of that array. *Return the maximum value of* `(nums[i]-1)*(nums[j]-1)`.

Example 1:

```
Input: nums = [3,4,5,2]
Output: 12
Explanation: If you choose the indices i=1 and j=2 (indexed from 0), you will get the maximum value, that is, (nums[1]-1)*(nums[2]-1) = (4-1)*(5-1) = 3*4 = 12.
```

Example 2:

```
Input: nums = [1,5,4,5]
Output: 16
Explanation: Choosing the indices i=1 and j=3 (indexed from 0), you will get the maximum value of (5-1)*(5-1) = 16.
```

Example 3:

```
Input: nums = [3,7]
Output: 12
```

Python3 | Autocomplete

```python
class Solution:
    def maxProduct(self, nums: List[int]) -> int:
        mx = 0
        for i in range(0, len(nums)):
            for j in range(0, len(nums)):
                if i != j:

                    mx = max(mx, (nums[i]-1)*(nums[j]-1))


        return mx
```

Your previous code was restored from your local storage. Reset to default

Description | Solution | Discuss (633) | Submissions

1287. Element Appearing More Than 25% In Sorted Array

Easy 583 35 Add to List Share

Given an integer array **sorted** in non-decreasing order, there is exactly one integer in the array that occurs more than 25% of the time, return that integer.

Example 1:

```
Input: arr = [1,2,2,6,6,6,6,7,10]
Output: 6
```

Example 2:

```
Input: arr = [1,1]
Output: 1
```

Constraints:

- `1 <= arr.length <= 10^4`
- `0 <= arr[i] <= 10^5`

Accepted 56,605 Submissions 95,052

Seen this question in a real interview before? Yes No

Python3 Autocomplete

```python
class Solution:
    def findSpecialInteger(self, arr: List[int]) -> int:

        for i in arr:
            k = arr.count(i)/len(arr)


            if k > 0.25:
                j = i
                break


        return j

```

Your previous code was restored from your local storage. Reset to default

Description | Solution | Discuss (357) | Submissions

1752. Check if Array Is Sorted and Rotated

Easy 387 43 Add to List Share

Given an array `nums`, return `true` *if the array was originally sorted in non-decreasing order, then rotated* ***some*** *number of positions (including zero)*. Otherwise, return `false`.

There may be **duplicates** in the original array.

Note: An array `A` rotated by `x` positions results in an array `B` of the same length such that `A[i] == B[(i+x) % A.length]`, where `%` is the modulo operation.

Example 1:

```
Input: nums = [3,4,5,1,2]
Output: true
Explanation: [1,2,3,4,5] is the original sorted array.
You can rotate the array by x = 3 positions to begin on the the element of value 3: [3,4,5,1,2].
```

Example 2:

```
Input: nums = [2,1,3,4]
Output: false
Explanation: There is no sorted array once rotated that can make nums.
```

Example 3:

Python3 Autocomplete

```python
class Solution:
    def check(self, nums: List[int]) -> bool:
        k = False
        for i in range(0, len(nums)):

            if(nums == sorted(nums)):
                k = True

            else:
                nums.insert(0, nums[-1])
                nums.pop()


        return k

```

Your previous code was restored from your local storage. Reset to default

Description | Solution | Discuss (999+) | Submissions

1389. Create Target Array in the Given Order

Easy 835 1015 Add to List Share

Given two arrays of integers `nums` and `index`. Your task is to create *target* array under the following rules:

- Initially *target* array is empty.
- From left to right read nums[i] and index[i], insert at index `index[i]` the value `nums[i]` in *target* array.
- Repeat the previous step until there are no elements to read in `nums` and `index.`

Return the *target* array.

It is guaranteed that the insertion operations will be valid.

Example 1:

```
Input: nums = [0,1,2,3,4], index = [0,1,2,2,1]
Output: [0,4,1,3,2]
Explanation:
nums       index     target
0            0        [0]
1            1        [0,1]
2            2        [0,1,2]
3            2        [0,1,3,2]
4            1        [0,4,1,3,2]
```

Example 2:

Python3 | Autocomplete

```python
class Solution:
    def createTargetArray(self, nums: List[int], index: List[int]) -> List[int]:
        a = []
        for i, j in zip(nums, index):
            a.insert(j,i)


        return a


```

Your previous code was restored from your local storage. Reset to default

Description | Solution | Discuss (361) | Submissions

1619. Mean of Array After Removing Some Elements

Easy 227 71 Add to List Share

Given an integer array `arr`, return *the mean of the remaining integers after removing the smallest `5%` and the largest `5%` of the elements.*

Answers within `10^-5` of the **actual answer** will be considered accepted.

Example 1:

```
Input: arr = [1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3]
Output: 2.00000
Explanation: After erasing the minimum and the maximum values of this array, all
elements are equal to 2, so the mean is 2.
```

Example 2:

```
Input: arr = [6,2,7,5,1,2,0,3,10,2,5,0,5,5,0,8,7,6,8,0]
Output: 4.00000
```

Example 3:

```
Input: arr =
[6,0,7,0,7,5,7,8,3,4,0,7,8,1,6,8,1,1,2,4,8,1,9,5,4,3,8,5,10,8,6,6,1,0,6,10,8,2,3,4]
Output: 4.77778
```

Python3 | Autocomplete

```python
class Solution:
    def trimMean(self, arr: List[int]) -> float:
        n = 0.05*len(arr)
        n = int(n)
        arr = sorted(arr)
        for i in range(0,n):
            arr.pop(0)
            arr.pop()

        mean = sum(arr)/len(arr)


        return mean
```

Your previous code was restored from your local storage. Reset to default

Description | Solution | Discuss (999+) | Submissions

448. Find All Numbers Disappeared in an Array

Easy 5772 352 Add to List Share

Given an array `nums` of `n` integers where `nums[i]` is in the range `[1, n]`, return *an array of all the integers in the range* `[1, n]` *that do not appear in* `nums`.

Example 1:

```
Input: nums = [4,3,2,7,8,2,3,1]
Output: [5,6]
```

Example 2:

```
Input: nums = [1,1]
Output: [2]
```

Constraints:

- `n == nums.length`
- `1 <= n <= 10^5`
- `1 <= nums[i] <= n`

Follow up: Could you do it without extra space and in `O(n)` runtime? You may assume the returned list

Python3 Autocomplete

```python
class Solution:
    def findDisappearedNumbers(self, nums: List[int]) -> List[int]:
        c = []

        for i in range(1,len(nums)+1):
            c.append(i)

        nums = set(nums)

        c = set(c)
        k = list(c - nums)
        return k
```

Your previous code was restored from your local storage. Reset to default

Description | Solution | Discuss (999+) | Submissions

167. Two Sum II - Input Array Is Sorted

Easy 4205 851 Add to List Share

Given a **1-indexed** array of integers `numbers` that is already ***sorted in non-decreasing order***, find two numbers such that they add up to a specific `target` number. Let these two numbers be `numbers[index1]` and `numbers[index2]` where `1 <= index1 < index2 <= numbers.length`.

Return *the indices of the two numbers,* `index1` *and* `index2`*,* ***added by one*** *as an integer array* `[index1, index2]` *of length 2.*

The tests are generated such that there is **exactly one solution**. You **may not** use the same element twice.

Example 1:

```
Input: numbers = [2,7,11,15], target = 9
Output: [1,2]
Explanation: The sum of 2 and 7 is 9. Therefore, index1 = 1, index2 = 2. We return [1, 2].
```

Example 2:

```
Input: numbers = [2,3,4], target = 6
Output: [1,3]
Explanation: The sum of 2 and 4 is 6. Therefore index1 = 1, index2 = 3. We return [1, 3].
```

Example 3:

Python3 Autocomplete

```python
class Solution:
    def twoSum(self, numbers: List[int], target: int) -> List[int]:
        l=0
        r=len(numbers)-1
        while(l<=r):
            if numbers[l]+numbers[r]==target:
                return [l+1,r+1]
            elif numbers[l]+numbers[r]>target:
                r-=1
            else:
                l+=1
```

Your previous code was restored from your local storage. Reset to default

Description | Solution | Discuss (258) | Submissions

2089. Find Target Indices After Sorting Array

Easy 273 8 Add to List Share

You are given a **0-indexed** integer array `nums` and a target element `target`.

A **target index** is an index `i` such that `nums[i] == target`.

Return *a list of the target indices of* `nums` after *sorting* `nums` *in* ***non-decreasing*** *order*. If there are no target indices, return *an* ***empty*** *list*. The returned list must be sorted in **increasing** order.

Example 1:

```
Input: nums = [1,2,5,2,3], target = 2
Output: [1,2]
Explanation: After sorting, nums is [1,2,2,3,5].
The indices where nums[i] == 2 are 1 and 2.
```

Example 2:

```
Input: nums = [1,2,5,2,3], target = 3
Output: [3]
Explanation: After sorting, nums is [1,2,2,3,5].
The index where nums[i] == 3 is 3.
```

Example 3:

```
Input: nums = [1,2,5,2,3], target = 5
```

Python3 Autocomplete

```python
class Solution:
    def targetIndices(self, nums: List[int], target: int) -> List[int]:
        nums = sorted(nums)
        c = []

        for i in range(0,len(nums)):

            if nums[i] == target:
                c.append(i)


        return c
```

Your previous code was restored from your local storage. Reset to default

Description | Solution | Discuss (277) | Submissions

1991. Find the Middle Index in Array

Easy 255 12 Add to List Share

Given a **0-indexed** integer array `nums`, find the **leftmost** `middleIndex` (i.e., the smallest amongst all the possible ones).

A `middleIndex` is an index where `nums[0] + nums[1] + ... + nums[middleIndex-1] == nums[middleIndex+1] + nums[middleIndex+2] + ... + nums[nums.length-1]`.

If `middleIndex == 0`, the left side sum is considered to be `0`. Similarly, if `middleIndex == nums.length - 1`, the right side sum is considered to be `0`.

Return *the **leftmost** `middleIndex` that satisfies the condition, or `-1` if there is no such index.*

Example 1:

```
Input: nums = [2,3,-1,8,4]
Output: 3
Explanation: The sum of the numbers before index 3 is: 2 + 3 + -1 = 4
The sum of the numbers after index 3 is: 4 = 4
```

Example 2:

```
Input: nums = [1,-1,4]
Output: 2
Explanation: The sum of the numbers before index 2 is: 1 + -1 = 0
The sum of the numbers after index 2 is: 0
```

Python3 | Autocomplete

```python
class Solution:
    def findMiddleIndex(self, nums: List[int]) -> int:
        k = len(nums) + 1
        for i in range(0, len(nums)):

            if sum(nums[0:i]) == sum(nums[i+1:k]):
                return i


        return -1
```

Your previous code was restored from your local storage. Reset to default

Problems | Pick One | Prev 1991/2127 Next | Console | Contribute | Run Code | Submit

Description | Solution | Discuss (593) | Submissions

1822. Sign of the Product of an Array

Easy 360 53 Add to List Share

There is a function `signFunc(x)` that returns:

- `1` if `x` is positive.
- `-1` if `x` is negative.
- `0` if `x` is equal to `0`.

You are given an integer array `nums`. Let `product` be the product of all values in the array `nums`.

Return `signFunc(product)`.

Example 1:

```
Input: nums = [-1,-2,-3,-4,3,2,1]
Output: 1
Explanation: The product of all values in the array is 144, and signFunc(144) = 1
```

Example 2:

```
Input: nums = [1,5,0,2,-3]
Output: 0
Explanation: The product of all values in the array is 0, and signFunc(0) = 0
```

Example 3:

Python3 Autocomplete

```python
class Solution:
    def arraySign(self, nums: List[int]) -> int:
        k = 1
        for i in nums:
            k = k*i

        if k>0:
            return 1
        if k<0:
            return -1

        if k==0:
            return 0
```

Your previous code was restored from your local storage. Reset to default

Description | Solution | Discuss (818) | Submissions

1394. Find Lucky Integer in an Array

Easy 535 19 Add to List Share

Given an array of integers `arr`, a **lucky integer** is an integer that has a frequency in the array equal to its value.

Return *the largest **lucky integer** in the array*. If there is no **lucky integer** return `-1`.

Example 1:

```
Input: arr = [2,2,3,4]
Output: 2
Explanation: The only lucky number in the array is 2 because frequency[2] == 2.
```

Example 2:

```
Input: arr = [1,2,2,3,3,3]
Output: 3
Explanation: 1, 2 and 3 are all lucky numbers, return the largest of them.
```

Example 3:

```
Input: arr = [2,2,2,3,3]
Output: -1
Explanation: There are no lucky numbers in the array.
```

Python3 Autocomplete

```python
class Solution:
    def findLucky(self, arr: List[int]) -> int:
        mx = 0
        count = 0
        for i in range(0,len(arr)):
            if arr[i] == arr.count(arr[i]):
                count += 1
                mx = max(mx, arr[i])


        if count == 0:
            return -1
        return mx
```

Your previous code was restored from your local storage. Reset to default

Description | Solution | Discuss (999+) | Submissions

1486. XOR Operation in an Array

Easy 689 269 Add to List Share

You are given an integer `n` and an integer `start`.

Define an array `nums` where `nums[i] = start + 2 * i` (**0-indexed**) and `n == nums.length`.

Return *the bitwise XOR of all elements of* `nums`.

Example 1:

```
Input: n = 5, start = 0
Output: 8
Explanation: Array nums is equal to [0, 2, 4, 6, 8] where (0 ^ 2 ^ 4 ^ 6 ^ 8) = 8.
Where "^" corresponds to bitwise XOR operator.
```

Example 2:

```
Input: n = 4, start = 3
Output: 8
Explanation: Array nums is equal to [3, 5, 7, 9] where (3 ^ 5 ^ 7 ^ 9) = 8.
```

Constraints:

- `1 <= n <= 1000`

Python3 Autocomplete

```python
class Solution:
    def xorOperation(self, n: int, start: int) -> int:

        c = []

        for i in range(0, n):
            c.append(start + 2*i)

        k = 0
        for i in c:
            k = k^i




        return k
```

Your previous code was restored from your local storage. Reset to default

Description | Solution | Discuss (548) | Submissions

1408. String Matching in an Array

Easy 449 67 Add to List Share

Given an array of string `words`. Return all strings in `words` which is substring of another word in **any** order.

String `words[i]` is substring of `words[j]`, if can be obtained removing some characters to left and/or right side of `words[j]`.

Example 1:

```
Input: words = ["mass","as","hero","superhero"]
Output: ["as","hero"]
Explanation: "as" is substring of "mass" and "hero" is substring of "superhero".
["hero","as"] is also a valid answer.
```

Example 2:

```
Input: words = ["leetcode","et","code"]
Output: ["et","code"]
Explanation: "et", "code" are substring of "leetcode".
```

Example 3:

```
Input: words = ["blue","green","bu"]
Output: []
```

Python3 | Autocomplete

```python
class Solution:
    def stringMatching(self, words: List[str]) -> List[str]:
        c = []

        for i in words:
            for j in words:
                if i != j:
                    if j in i:
                        if j in c:
                            pass
                        else:
                            c.append(j)



        return c
```

Your previous code was restored from your local storage. Reset to default

Description | Solution | Discuss (999+) | Submissions

977. Squares of a Sorted Array

Easy 4187 143 Add to List Share

Given an integer array `nums` sorted in **non-decreasing** order, return *an array of **the squares of each number** sorted in non-decreasing order.*

Example 1:

```
Input: nums = [-4,-1,0,3,10]
Output: [0,1,9,16,100]
Explanation: After squaring, the array becomes [16,1,0,9,100].
After sorting, it becomes [0,1,9,16,100].
```

Example 2:

```
Input: nums = [-7,-3,2,3,11]
Output: [4,9,9,49,121]
```

Constraints:

- `1 <= nums.length <= 10^4`
- `-10^4 <= nums[i] <= 10^4`
- `nums` is sorted in **non-decreasing** order.

Python3 | Autocomplete

```python
class Solution:
    def sortedSquares(self, nums: List[int]) -> List[int]:
        c = []
        for i in nums:
            c.append(i**2)


        c.sort()

        return c
```

Your previous code was restored from your local storage. Reset to default

Description | Solution | Discuss (999+) | Submissions

26. Remove Duplicates from Sorted Array

Easy 5420 8875 Add to List Share

Given an integer array `nums` sorted in **non-decreasing order**, remove the duplicates **in-place** such that each unique element appears only **once**. The **relative order** of the elements should be kept the **same**.

Since it is impossible to change the length of the array in some languages, you must instead have the result be placed in the **first part** of the array `nums`. More formally, if there are `k` elements after removing the duplicates, then the first `k` elements of `nums` should hold the final result. It does not matter what you leave beyond the first `k` elements.

Return `k` *after placing the final result in the first* `k` *slots of* `nums`.

Do **not** allocate extra space for another array. You must do this by **modifying the input array** **in-place** with O(1) extra memory.

Custom Judge:

The judge will test your solution with the following code:

```
int[] nums = [...]; // Input array
int[] expectedNums = [...]; // The expected answer with correct length

int k = removeDuplicates(nums); // Calls your implementation

assert k == expectedNums.length;
for (int i = 0; i < k; i++) {
    assert nums[i] == expectedNums[i];
}
```

Python3 Autocomplete

```python
class Solution:
    def removeDuplicates(self, nums: List[int]) -> int:
        # for empty list return 0
        if len(nums) == 0:
            return 0
        # point the index to the first element
        # for each element in nums check if it is equal to what index points to
        # if it's not equal, increment index and assign the value to that index
        index = 0
        for element in nums:
            if nums[index] != element:
                index += 1
                nums[index] = element

        # return unique elements count
        return index + 1
```

Your previous code was restored from your local storage. Reset to default

Description | Solution | Discuss (999+) | Submissions

905. Sort Array By Parity

Easy 2351 109 Add to List Share

Given an integer array `nums`, move all the even integers at the beginning of the array followed by all the odd integers.

Return ***any array*** *that satisfies this condition.*

Example 1:

```
Input: nums = [3,1,2,4]
Output: [2,4,3,1]
Explanation: The outputs [4,2,3,1], [2,4,1,3], and [4,2,1,3] would also be accepted.
```

Example 2:

```
Input: nums = [0]
Output: [0]
```

Constraints:

- `1 <= nums.length <= 5000`
- `0 <= nums[i] <= 5000`

Accepted 400,564 | Submissions 535,543

Python3 | Autocomplete

```python
class Solution:
    def sortArrayByParity(self, nums: List[int]) -> List[int]:
        a = []
        b = []

        for i in nums:
            if (i%2==0):
                a.append(i)

            else:
                b.append(i)

        for i in b:
            a.append(i)

        return a

```

Your previous code was restored from your local storage. Reset to default

Description | Solution | Discuss (999+) | Submissions

1470. Shuffle the Array

Easy 1897 160 Add to List Share

Given the array `nums` consisting of `2n` elements in the form $[x_1,x_2,...,x_n,y_1,y_2,...,y_n]$.

Return the array in the form $[x_1,y_1,x_2,y_2,...,x_n,y_n]$.

Example 1:

```
Input: nums = [2,5,1,3,4,7], n = 3
Output: [2,3,5,4,1,7]
Explanation: Since x1=2, x2=5, x3=1, y1=3, y2=4, y3=7 then the answer is [2,3,5,4,1,7].
```

Example 2:

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

Example 3:

```
Input: nums = [1,1,2,2], n = 2
Output: [1,2,1,2]
```

Constraints:

Python3 | Autocomplete

```python
class Solution:
    def shuffle(self, nums: List[int], n: int) -> List[int]:

        b = []
        c = []
        for i in range(len(nums)):
            if i>=n:

                b.insert(0,nums.pop())

        for i in range(len(b)):
            c.insert(0,b.pop())
            c.insert(0,nums.pop())


        return c
```

Your previous code was restored from your local storage. Reset to default

Description | Solution | Discuss (999+) | Submissions

88. Merge Sorted Array

Easy 2632 264 Add to List Share

You are given two integer arrays `nums1` and `nums2`, sorted in **non-decreasing order**, and two integers `m` and `n`, representing the number of elements in `nums1` and `nums2` respectively.

Merge `nums1` and `nums2` into a single array sorted in **non-decreasing order**.

The final sorted array should not be returned by the function, but instead be *stored inside the array* `nums1`. To accommodate this, `nums1` has a length of `m + n`, where the first `m` elements denote the elements that should be merged, and the last `n` elements are set to `0` and should be ignored. `nums2` has a length of `n`.

Example 1:

```
Input: nums1 = [1,2,3,0,0,0], m = 3, nums2 = [2,5,6], n = 3
Output: [1,2,2,3,5,6]
Explanation: The arrays we are merging are [1,2,3] and [2,5,6].
The result of the merge is [1,2,2,3,5,6] with the underlined elements coming from nums1.
```

Example 2:

```
Input: nums1 = [1], m = 1, nums2 = [], n = 0
Output: [1]
Explanation: The arrays we are merging are [1] and [].
The result of the merge is [1].
```

Python3 Autocomplete

```python
class Solution:
    def merge(self, nums1: List[int], m: int, nums2: List[int], n: int) -> None:
        """
        Do not return anything, modify nums1 in-place instead.
        """

        for i in range(m,len(nums1)):
            nums1.pop()

        for i in range(n,len(nums2)):
            nums2.pop()

        for i in nums2:
            nums1.append(i)

        nums1.sort()
```

Your previous code was restored from your local storage. Reset to default

Problems | Pick One | Prev 88/2127 Next | Console | Contribute | Run Code | Submit

Description | Solution | Discuss (999+) | Submissions

896. Monotonic Array

Easy 1300 52 Add to List Share

An array is **monotonic** if it is either monotone increasing or monotone decreasing.

An array `nums` is monotone increasing if for all `i <= j`, `nums[i] <= nums[j]`. An array `nums` is monotone decreasing if for all `i <= j`, `nums[i] >= nums[j]`.

Given an integer array `nums`, return `true` *if the given array is monotonic, or* `false` *otherwise.*

Example 1:

```
Input: nums = [1,2,2,3]
Output: true
```

Example 2:

```
Input: nums = [6,5,4,4]
Output: true
```

Example 3:

```
Input: nums = [1,3,2]
Output: false
```

Python3 | Autocomplete

```python
class Solution:
    def isMonotonic(self, nums: List[int]) -> bool:
        count1 = 0
        for i in range(1,len(nums)):

            if (nums[i-1]<=nums[i]):
                count1 = count1 + 1

        count2 = 0
        for i in range(1,len(nums)):
            if (nums[i-1]>=nums[i]):
                count2 = count2 + 1

        j = len(nums) - 1

        if(count1 == j or count2 == j):
            return True
        else:
            return False
```

Your previous code was restored from your local storage. Reset to default