



Subscribe to see which companies asked this question.

Related Topics ▾

Similar Questions ▾

Py3



```
1 class Solution:
2     def PredictTheWinner(self,
3         nums: List[int]) -> bool:
4         if len(nums)<=1:
5             return False
6         k=sum(nums)
7         if k%3==0:
8             return True
9         return False
```





Example 1:

Input: nums = [1, 5, 2]

Output: false

Explanation: Initially, player 1 can choose either 1 or 5. If he chooses 5, then player 2 will choose 1 because 5 is odd. So, final score of player 1 is 1 + 2 = 3 and player 2's score is 5. Hence, player 1 will never be the winner.

Example 2:

Input: nums = [1, 5, 233, 7]

Output: true

Explanation: Player 1 first chooses 1. Finally, player 1 has more score (234).

Constraints:

- $1 \leq \text{nums.length} \leq 20$
- $0 \leq \text{nums}[i] \leq 10^7$

Seen this question in a real interview before?

Yes

No

Subscribe to see which companies asked this question.

Related Topics ▾





486. Predict the Winner

Description

Hints

Submissions

Discuss

Pick One

You are given an integer array `nums`. Two players are playing a game with this array: player 1 and player 2.

Player 1 and player 2 take turns, with player 1 starting first. Both players start the game with a score of `0`. At each turn, the player takes one of the numbers from either end of the array (i.e., `nums[0]` or `nums[nums.length - 1]`) which reduces the size of the array by `1`. The player adds the chosen number to their score. The game ends when there are no more elements in the array.

Return `true` if Player 1 can win the game. If the scores of both players are equal, then player 1 is still the winner, and you should also return `true`. You may assume that both players are playing optimally.



- `i == nums.length`
- `1 <= nums.length <= 105`
- `-109 <= nums[i] <= 109`

Seen this question in a real interview before?

Yes

No

Subscribe to see which companies asked this question.

Related Topics ▾

Similar Questions ▾

Py3



```
1 class Solution:
2     def minMoves2(self, nums: List[int]) -> int:
3         nums.sort()
4         m=nums[len(nums)//2]
5         move=0
6         for i in nums:
7             move=move+abs(i-m)
8         return move
9 
```





462. Minimum Moves to Equal Array Elements II

Description

Hints

Submissions

Discuss

Pick One

Given an integer array `nums` of size `n`, return the minimum number of moves required to make all array elements equal.

In one move, you can increment or decrement an element of the array by `1`.

Test cases are designed so that the answer will fit in a **32-bit** integer.

Example 1:

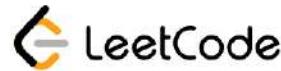
Input: `nums = [1, 2, 3]`

Output: `2`

Explanation:

Only two moves are needed (remember each move changes exactly one element).
`[1, 2, 3] => [2, 2, 3] => [2, 2, 2]`

Example 2:



[Subscribe](#) to see which companies asked this question.

[Related Topics ▾](#)

Py3



```
1 class Solution:
2     def findContentChildren(self,
g: List[int], s: List[int]) ->
int:
3         k=len(g)
4         m=len(s)
5         c=0
6         g.sort()
7         s.sort()
8         i=0
9         j=0
10        while i<k and j<m:
11            if s[j]>=g[i]:
12                c=c+1
13                i=i+1
14                j=j+1
15        return c
16
```





455. Assign Cookies

[Description](#)[Hints](#)[Submissions](#)[Discuss](#)[Pick One](#)

Assume you are an awesome parent and want to give your children some cookies. But, you should give each child at most one cookie.

Each child i has a greed factor $g[i]$, which is the minimum size of a cookie that the child will be content with; and each cookie j has a size $s[j]$. If $s[j] \geq g[i]$, we can assign the cookie j to the child i , and the child i will be content. Your goal is to maximize the number of your content children and output the maximum number.

Example 1:

Input: $g = [1, 2, 3]$, $s = [1, 1]$

Output: 1

Explanation: You have 3 children and 2 And even though you have 2 cookies, si You need to output 1.

Example 2:





LeetCode

Related topics

Similar Questions ▾

Py3



```
1 from collections import
2 defaultdict
3 class Solution:
4     def fourSumCount(self, nums1: List[int], nums2: List[int],
5         nums3: List[int], nums4: List[int]) -> int:
6         sums=defaultdict(int)
7         c=0
8         for i in nums1:
9             for j in nums2:
10                sums[i+j] += 1
11         for k in nums3:
12             for h in nums4:
13                 c=c+sums[-(k+h)]
14
15         return c
```



Custom Testcase ([Contribute !](#))



454. 4Sum II

[Description](#)[Hints](#)[Submissions](#)[Discuss](#)[Pick One](#)

Given four integer arrays `nums1`, `nums2`, `nums3`, and `nums4` all of length `n`, return the number of tuples `(i, j, k, l)` such that:

- `0 <= i, j, k, l < n`
- `nums1[i] + nums2[j] + nums3[k] + nums4[l] == 0`

Example 1:

```
-1, 2], nums4 = [0, 2]
```

```
] + nums4[1] = 1 + (-2) + (-1) + 2 = 0
] + nums4[0] = 2 + (-1) + (-1) + 0 = 0
```

Example 2:

```
Input: nums1 = [0], nums2 = [0], nums3 = [0]
Output: 1
```





454. 4Sum II

[Description](#)[Hints](#)[Submissions](#)[Discuss](#)[Pick One](#)

Given four integer arrays `nums1`, `nums2`, `nums3`, and `nums4` all of length `n`, return the number of tuples `(i, j, k, l)` such that:

- $0 \leq i, j, k, l < n$
- $\text{nums1}[i] + \text{nums2}[j] + \text{nums3}[k] + \text{nums4}[l] = 0$

Example 1:

`-2, -1], nums3 = [-1, 2], nums4 = [0, 2]`

`ums2[0] + nums3[0] + nums4[1] = 1 + (-2)`
`ums2[1] + nums3[0] + nums4[0] = 2 + (-1)`

Example 2:

Input: `nums1 = [0], nums2 = [0], nums3 = [0], nums4 = [0]`
Output: 1





454. 4Sum II

[Description](#)[Hints](#)[Submissions](#)[Discuss](#)[Pick One](#)

Given four integer arrays `nums1`, `nums2`, `nums3`, and `nums4` all of length `n`, return the number of tuples `(i, j, k, l)` such that:

- $0 \leq i, j, k, l < n$
- $\text{nums1}[i] + \text{nums2}[j] + \text{nums3}[k] + \text{nums4}[l] = 0$

Example 1:

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

Output: 2

Explanation:

The two tuples are:

1. $(0, 0, 0, 1) \rightarrow \text{nums1}[0] + \text{nums2}[0]$

2. $(1, 1, 0, 0) \rightarrow \text{nums1}[1] + \text{nums2}[1]$

Example 2:

Input: `nums1 = [0]`, `nums2 = [0]`, `nums3 = [0]`

Output: 1





- The answer is guaranteed to fit in a **32-bit** integer.

Seen this question in a real interview before?

Yes

No

Subscribe to see which companies asked this question.

Related Topics ▾

Similar Questions ▾

Py3



```
1 class Solution:  
2     def minMoves(self, nums:  
3         List[int]) -> int:  
4             n=len(nums)  
5             g=min(nums)  
6             tc=sum(nums)  
7             f=(tc-g*n)  
8             return f  
9
```





453. Minimum Moves to Equal Array Elements

Description

Hints

Submissions

Discuss

Pick One

Given an integer array `nums` of size `n`, return the minimum number of moves required to make all array elements equal.

In one move, you can increment `n - 1` elements of the array by `1`.

Example 1:

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

Output: `3`

Explanation: Only three moves are needed.
`[1, 2, 3] => [2, 3, 3] => [3, 4, 3] =>`

Example 2:

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





Follow up: Could you do it without extra space and in $O(n)$ runtime? You may assume the returned list does not count as extra space.

Seen this question in a real interview before?



Yes No

No

Subscribe to see which companies asked this question.

Related Topics ▾

Similar Questions ▾

Py3





448. Find All Numbers Disappeared in an Array

Description

Hints

Submissions

Discuss

Pick One

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:





Follow up: Can you find an $O(n)$ solution?

Seen this question in a real interview before?

Yes

No

Subscribe to see which companies asked this question.

Related Topics ▾

Similar Questions ▾

Py3



```
1 class Solution:
2     def thirdMax(self, nums: List[int]) -> int:
3         k=list(set(nums))
4         if len(k)<3:
5             return max(k)
6         k.remove(max(k))
7         k.remove(max(k))
8         return max(k)
9
```





414. Third Maximum Number

Description

Hints

Submissions

Discuss

Pick One

Given an integer array `nums`, return the **third distinct maximum number** in this array. If the third maximum does not exist, return the **maximum number**.

Example 1:

Input: nums = [3, 2, 1]

Output: 1

Explanation:

The first distinct maximum is 3.

The second distinct maximum is 2.

The third distinct maximum is 1.

Example 2:

Input: nums = [1, 2]

Output: 2

Explanation:

The first distinct maximum is 2.





```
1 class Solution:
2     def
3         numberofArithmeticSlices(self,
4             nums: List[int]) -> int:
5                 if len(nums)<3:
6                     return 0
7                     l=[]
8                     c=0
9                     s=0
10
11                     k=len(nums)
12                     for i in range(2,k):
13                         if nums[i]-nums[i-
14                             1]==nums[i-1]-nums[i-2]:
15                             s=s+1
16                             c=c+s
17                         else:
18                             s=0
19
20
21
```

Custom Testcase ([Contribute](#))

Run

Submit



413. Arithmetic Slices

[Description](#)[Hints](#)[Submissions](#)[Discuss](#)[Pick One](#)

An integer array is called arithmetic if it consists of **at least three elements** and if the difference between any two consecutive elements is the same.

- For example, `[1, 3, 5, 7, 9]`, `[7, 7, 7, 7]`, and `[3, -1, -5, -9]` are arithmetic sequences.

Given an integer array `nums`, return *the number of arithmetic subarrays of `nums`*.

A **subarray** is a contiguous subsequence of the array.

Example 1:

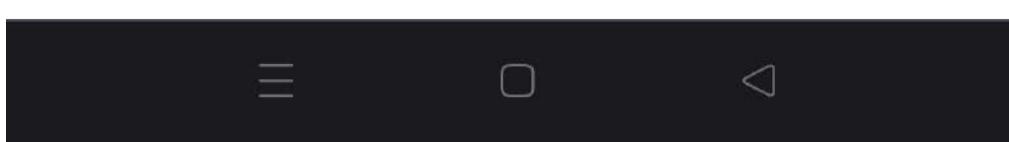
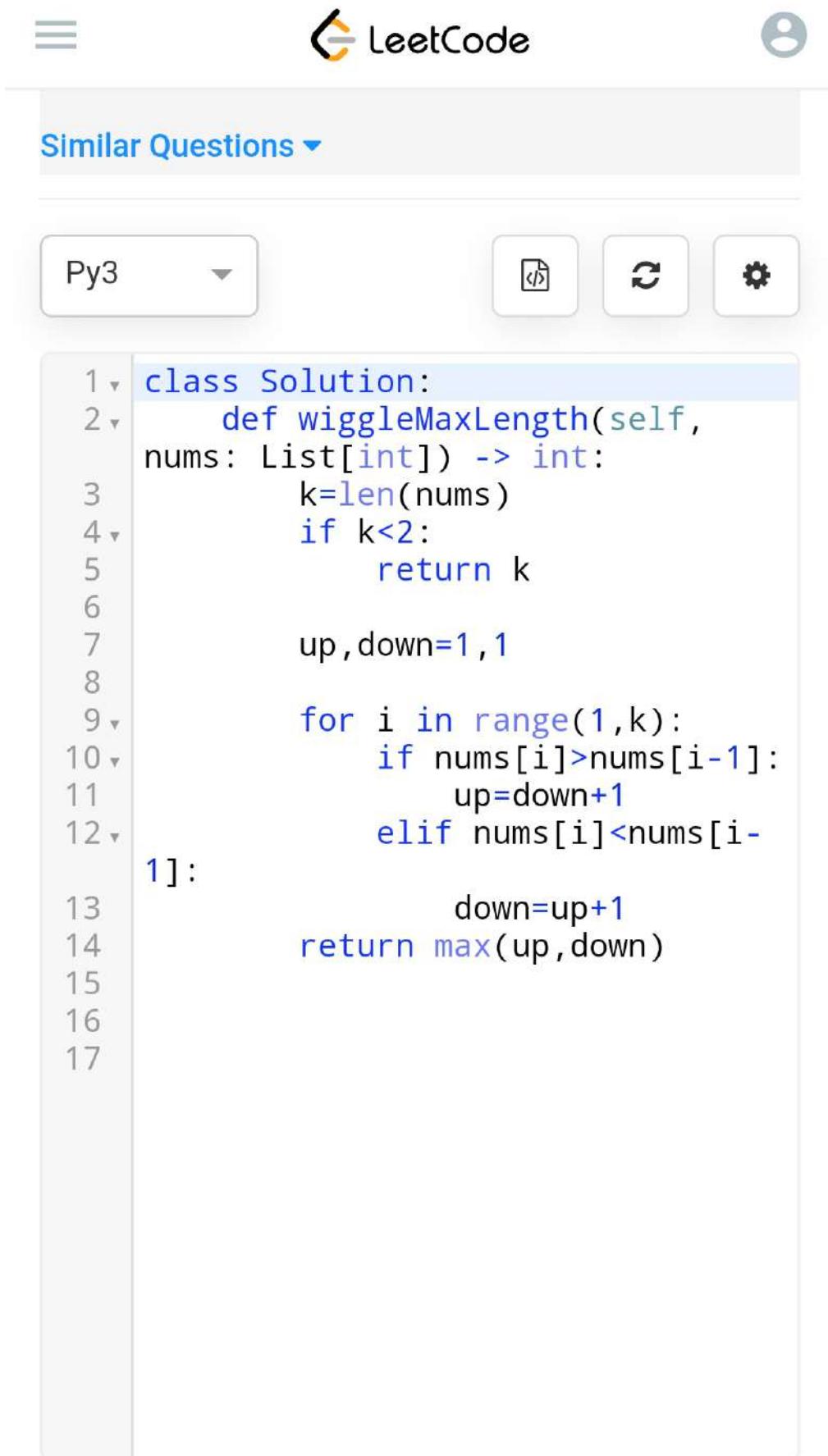
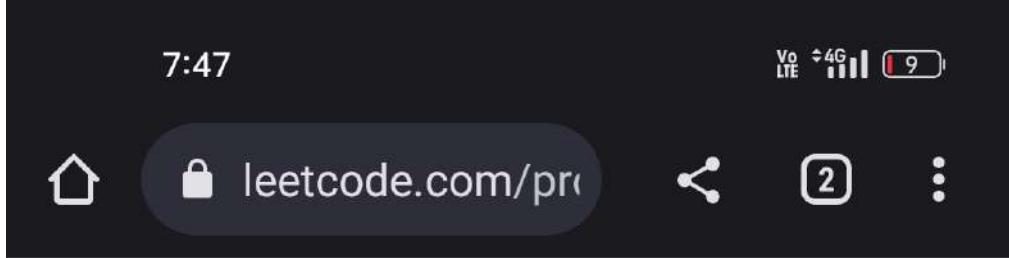
Input: `nums = [1, 2, 3, 4]`

Output: 3

Explanation: We have 3 arithmetic slices:

Example 2:







leaving the remaining elements in their original order.

Given an integer array `nums`, return the length of the longest **wiggle subsequence** of `nums`.

Example 1:

```
t: nums = [1, 7, 4, 9, 2, 5]
ut: 6
anation: The entire sequence is a wiggle
```

Example 2:

```
t: nums = [1, 17, 5, 10, 13, 15, 10, 5, 16, 8]
ut: 7
anation: There are several subsequences
is [1, 17, 10, 13, 10, 16, 8] with diff
```

Example 3:

```
Input: nums = [1, 2, 3, 4, 5, 6, 7, 8, 9]
Output: 2
```

Constraints:

- `1 <= nums.length <= 1000`
- `0 <= nums[i] <= 1000`





376. Wiggle Subsequence

Description

Hints

Submissions

Discuss

Pick One

A **wiggle sequence** is a sequence where the differences between successive numbers strictly alternate between positive and negative. The first difference (if one exists) may be either positive or negative. A sequence with one element and a sequence with two non-equal elements are trivially wiggle sequences.

- For example, [1, 7, 4, 9, 2, 5] is a **wiggle sequence** because the differences (6, -3, 5, -7, 3) alternate between positive and negative.
- In contrast, [1, 4, 7, 2, 5] and [1, 7, 4, 5, 5] are not wiggle sequences. The first is not because its first two differences are positive, and the second is not because its last difference is zero.

A **subsequence** is obtained by deleting some





Subscribe to see which companies asked this question.

Related Topics ▾

Similar Questions ▾

Py3



```
1 from collections import Counter
2 class Solution:
3     def topKFrequent(self, nums: List[int], k: int) -> List[int]:
4         g=Counter(nums)
5         m = g.most_common(k)
6         l=[]
7         for h in m:
8             l.append(h[0])
9         return l
10
11
12
13
```





15233

538



347. Top K Frequent Elements

Description

Hints

Submissions

Discuss

Pick One

Given an integer array `nums` and an integer `k`, return the `k` most frequent elements. You may return the answer in any order.

Example 1:

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

Example 2:

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

Constraints:

1 <= nums.length <= 10⁵





question.

[Related Topics ▾](#)

[Similar Questions ▾](#)

Py3



```
1 class Solution:
2     def moveZeroes(self, nums: List[int]) -> None:
3         """
4             Do not return anything,
5             modify nums in-place instead.
6             """
7         return
8     nums.sort(key=lambda x:x==0)
```





14430

369



283. Move Zeroes

Description

Hints

Submissions

Discuss

Pick One

Given an integer array `nums`, move all `0`'s to the end of it while maintaining the relative order of the non-zero elements.

Note that you must do this in-place without making a copy of the array.

Example 1:

Input: `nums = [0, 1, 0, 3, 12]`

Output: `[1, 3, 12, 0, 0]`

Example 2:

Input: `nums = [0]`

Output: `[0]`





Py3



```
1 class Solution:
2     def hIndex(self, citations:
3         List[int]) -> int:
4             citations.sort(reverse=True)
5             k=len(citations)
6             for i in range(k):
7                 if citations [i]<i+1:
8                     return i
9             return k
```

Custom Testcase ([Contribute !](#))



times.

You must write an algorithm that runs in logarithmic time.

Example 1:

Input: citations = [0,1,3,5,6]

Output: 3

Explanation: [0,1,3,5,6] means the researcher has 5 papers with citation counts 0, 1, 3, 5 and 6. Since the researcher has 3 papers with

Example 2:

Input: citations = [1,2,100]

Output: 2

Constraints:

- n == citations.length
- 1 <= n <= 10⁵
- 0 <= citations[i] <= 1000
- citations is sorted in ascending order.

Seen this question in a real interview before?



Yes

No





275. H-Index II

Description

Hints

Submissions

Discuss

Pick One

Given an array of integers `citations` where
`citations[i]` is the number of citations a researcher received for their i^{th} paper and
`citations` is sorted in **ascending order**, return *the researcher's h-index*.

According to the [definition of h-index on Wikipedia](#):

The h-index is defined as the maximum value of `h` such that the given researcher has published at least `h` papers that have each been cited at least `h` times.

You must write an algorithm that runs in logarithmic time.

Example 1:

Input: `citations = [0, 1, 3, 5, 6]`

Output: 3





Similar Questions ▾

Py3



```
1 class Solution:
2     def hIndex(self, citations: List[int]) -> int:
3
4         citations.sort(reverse=True)
5         k=len(citations)
6         for i in range(k):
7             if citations[i]<i+1:
8                 return i
9
10
11
12
```



Example 1:

Input: citations = [3, 0, 6, 1, 5]

Output: 3

Explanation: [3, 0, 6, 1, 5] means the res

Since the researcher has 3 papers with

Example 2:

Input: citations = [1, 3, 1]

Output: 1

Constraints:

- n == citations.length
- 1 <= n <= 5000
- 0 <= citations[i] <= 1000

Seen this question in a real interview before?



Yes

No

Subscribe to see which companies asked this question.





274. H-Index

Description

Hints

Submissions

Discuss

Pick One

Given an array of integers `citations` where `citations[i]` is the number of citations a researcher received for their i^{th} paper, return *the researcher's h-index*.

According to the [definition of h-index on Wikipedia](#):
The h-index is defined as the maximum value of `h` such that the given researcher has published at least `h` papers that have each been cited at least `h` times.

Example 1:

Input: `citations = [3, 0, 6, 1, 5]`

Output: 3

Explanation: `[3, 0, 6, 1, 5]` means the researcher has 3 papers with

Example 2:

Input: `citations = [1, 3, 1]`





Subscribe to see which companies asked this question.

Related Topics ▾

Similar Questions ▾

Py3



```
1 class Solution:
2     def missingNumber(self, nums:
List[int]) -> int:
3         nums.sort()
4         k=len(nums)
5         s=k*(k+1)//2
6         a=sum(nums)
7         h=s-a
8         return h
```



268. Missing Number

Description

Hints

Submissions

Discuss

Pick One

Given an array `nums` containing `n` distinct numbers in the range `[0, n]`, return *the only number in the range that is missing from the array*.

Example 1:

Input: `nums = [3, 0, 1]`

Output: 2

Explanation: n = 3 since there are 3 r

Example 2:

Input: `nums = [0, 1]`

Output: 2

Explanation: n = 2 since there are 2 r

Example 3:





Py3



```
1 from collections import Counter
2
3 class Solution:
4     def containsDuplicate(self,
5         nums: List[int]) -> bool:
6         k=Counter(nums)
7         h=list(k.values())
8         for i in range(len(h)):
9             if h[i]>=2:
10                 return True
11
12         return False
```

Custom Testcase ([Contribute](#))



217. Contains Duplicate

Description

Hints

Submissions

Discuss

Pick One

Given an integer array `nums`, return `true` if any value appears **at least twice** in the array, and return `false` if every element is distinct.

Example 1:

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

Output: `true`

Example 2:

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

Output: `false`

Example 3:



Subscribe to see which companies asked this question.

[Related Topics ▾](#)

[Similar Questions ▾](#)

Py3



```
1 class Solution:  
2     def findKthLargest(self,  
3         nums: List[int], k: int) -> int:  
4             nums.sort()  
5             return nums[-k]
```





215. Kth Largest Element in an Array

Description

Hints

Submissions

Discuss

Pick One

Given an integer array `nums` and an integer `k`, return the `kth` largest element in the array.

Note that it is the `kth` largest element in the sorted order, not the `kth` distinct element.

Can you solve it without sorting?

Example 1:

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

Example 2:

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



Py3



```
1 from collections import Counter
2 class Solution:
3     def majorityElement(self,
4         nums: List[int]) -> int:
5         k=Counter(nums)
6         return k.most_common(1)[0][0]
```

Custom Testcase ([Contribute !](#))



10803

401



169. Majority Element

Description

Hints

Submissions

Discuss

Pick One

Given an array `nums` of size `n`, return *the majority element*.

The majority element is the element that appears more than $\lfloor n / 2 \rfloor$ times. You may assume that the majority element always exists in the array.

Example 1:

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

Example 2:

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





Similar Questions ·

Py3



```
1 class Solution:
2     def maximumGap(self, nums:
3         List[int]) -> int:
4             k=len(nums)
5             if k<=1:
6                 return 0
7             ma=0
8             nums.sort()
9             for i in range(1,k):
10                 m=nums[i]-nums[i-1]
11                 ma=max(ma,m)
12
13             return ma
14
```



Custom Testcase (Contribute !)





164. Maximum Gap

Description

Hints

Submissions

Discuss

Pick One

Given an integer array `nums`, return *the maximum difference between two successive elements in its sorted form*. If the array contains less than two elements, return `0`.

You must write an algorithm that runs in linear time and uses linear extra space.

Example 1:

Input: `nums = [3, 6, 9, 1]`

Output: `3`

Explanation: The sorted form of the ar

Example 2:

Input: `nums = [10]`

Output: `0`

Explanation: The array contains less t





Seen this question in a earlier view before?

Yes No

Subscribe to see which companies asked this question.

Related Topics ▾

Similar Questions ▾

Py3



```
1 class Solution:
2     def findPeakElement(self,
3         nums: List[int]) -> int:
4         lo=0
5         hi=len(nums)-1
6         while lo<hi:
7             mid=(lo+hi)//2
8             if
9                 nums[mid]>nums[mid+1]:
10                    hi= mid
11             else:
12                 lo=mid+1
13         return lo
```





Description



Hints



Submissions



Discuss

✖ Pick One

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 y

[Related Topics ▾](#)[Similar Questions ▾](#)

Py3



```
1 class Solution:
2     def findMin(self, nums:
3         List[int]) -> int:
4             k=float('inf')
5             for i in nums:
6                 if i<k:
7                     k=i
8             return k
```





much as possible.

Example 1:

Input: nums = [1, 3, 5]

Output: 1

Example 2:

Input: nums = [2, 2, 2, 0, 1]

Output: 0

Constraints:

- n == nums.length
- 1 <= n <= 5000
- -5000 <= nums[i] <= 5000
- nums is sorted and rotated between 1 and n times.

Follow up: This problem is similar to [Find Minimum in Rotated Sorted Array](#), but nums may contain **duplicates**. Would this affect the runtime?





154. Find Minimum in Rotated Sorted Array II

Description

Hints

Submissions

Discuss

Pick One

Suppose an array of length n sorted in ascending order is **rotated** between 1 and n times. For example, the array `nums = [0, 1, 4, 4, 5, 6, 7]` might become:

- `[4, 5, 6, 7, 0, 1, 4]` if it was rotated 4 times.
- `[0, 1, 4, 4, 5, 6, 7]` if it was rotated 7 times.

Notice that **rotating** an array `[a[0], a[1], a[2], ..., a[n-1]]` 1 time results in the array `[a[n-1], a[0], a[1], a[2], ..., a[n-2]]`.

Given the sorted rotated array `nums` that may contain **duplicates**, return *the minimum element of this array*.



Subscribe to see which companies asked this question.

Related Topics ▾

Similar Questions ▾

Py3



```
1 class Solution:
2     def findMin(self, nums:
3         List[int]) -> int:
4             k=float('inf')
5             for i in nums:
6                 if i<k:
7                     k=i
8             return k
```





time.

Example 1:

array was [1, 2, 3, 4, 5] rotated 3 times.

Example 2:

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

Output: 0

Explanation: The original array was [0, 1, 2, 3, 4, 5, 6]. After rotating by 4 times, it becomes [4, 5, 6, 7, 0, 1, 2].

Example 3:

Input: nums = [11, 13, 15, 17]

Output: 11

Explanation: The original array was [15, 17, 11, 13]. After rotating by 1 time, it becomes [11, 13, 15, 17].

Constraints:

- n == nums.length
- 1 <= n <= 5000
- -5000 <= nums[i] <= 5000





△ □ .

Given the sorted rotated array `nums` of unique elements, return *the minimum element of this array*.

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

Example 1:

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

Output: 1

Explanation: The original array was [1, 2, 3]

Example 2:

Input: `nums = [4, 5, 6, 7, 0, 1, 2]`

Output: 0

Explanation: The original array was [0, 1, 2, 3, 4, 5, 6, 7]

Example 3:

Input: `nums = [11, 13, 15, 17]`

Output: 11

Explanation: The original array was [11, 13, 15, 17, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Constraints:





153. Find Minimum in Rotated Sorted Array

[Description](#)[Hints](#)[Submissions](#)[Discuss](#)[Pick One](#)

Suppose an array of length n sorted in ascending order is **rotated** between 1 and n times. For example, the array `nums = [0, 1, 2, 4, 5, 6, 7]` might become:

- `[4, 5, 6, 7, 0, 1, 2]` if it was rotated 4 times.
- `[0, 1, 2, 4, 5, 6, 7]` if it was rotated 7 times.

Notice that **rotating** an array `[a[0], a[1], a[2], ..., a[n-1]]` 1 time results in the array `[a[n-1], a[0], a[1], a[2], ..., a[n-2]]`.

Given the sorted rotated array `nums` of **unique** elements, return *the minimum element of this array*.

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



[Related Topics ▾](#)[Similar Questions ▾](#)

Py3



```
1 class Solution:
2     def evalRPN(self, tokens: List[str]) -> int:
3         stack=[]
4         op={'+', '-', '*', '/'}
5         for i in tokens:
6             if i not in op:
7                 stack.append(int(i))
8             else:
9                 n2=stack.pop()
10                n1=stack.pop()
11                if i=='+':
12                    res=n1+n2
13                elif i=='-':
14                    res=n1-n2
15                elif i=='*':
16                    res=n1*n2
17                else:
18
19                res=int(n1/n2)
20                stack.append(res)
21
22
23
```

Custom Testcase ([Contribute](#) !)



expression in a reverse polish notation.

- The answer and all the intermediate calculations can be represented in a **32-bit** integer.

Example 1:

```
Input: tokens = [ "2" , "1" , "+" , "3" , "*" ]
```

```
Output: 9
```

```
Explanation: ((2 + 1) * 3) = 9
```

Example 2:

```
Input: tokens = [ "4" , "13" , "5" , "/" , "+" ]
```

```
Output: 6
```

```
Explanation: (4 + (13 / 5)) = 6
```

Example 3:

```
Input: tokens = [ "10" , "6" , "9" , "3" , "+" ,
```

```
Output: 22
```

```
Explanation: ((10 * (6 / ((9 + 3) * -1))) + 17) + 5  
= ((10 * (6 / (12 * -11))) + 17) + 5  
= ((10 * (6 / -132)) + 17) + 5  
= ((10 * 0) + 17) + 5  
= (0 + 17) + 5  
= 17 + 5  
= 22
```





150. Evaluate Reverse Polish Notation

Description



Hints



Submissions



Discuss

Pick One

You are given an array of strings `tokens` that represents an arithmetic expression in a [Reverse Polish Notation](#).

Evaluate the expression. Return an *integer* that represents the value of the expression.

Note that:

- The valid operators are `'+'`, `'-'`, `'*'`, and `'/'`.
- Each operand may be an integer or another expression.
- The division between two integers always **truncates toward zero**.
- There will not be any division by zero.
- The input represents a valid arithmetic expression in a reverse polish notation.
- The answer and all the intermediate





Seen this question in a real interview before?

 Yes No

Subscribe to see which companies asked this question.

Related Topics ▾**Similar Questions ▾**

Py3



```
1 from collections import Counter
2
3 class Solution:
4     def singleNumber(self, nums:
List[int]) -> int:
5         k=Counter(nums)
6         for num,c in k.items():
7             if c==1:
8                 return num
9
```





137. Single Number II

Description

Hints

Submissions

Discuss

Pick One

Given an integer array `nums` where every element appears **three times** except for one, which appears **exactly once**. *Find the single element and return it.*

You must implement a solution with a linear runtime complexity and use only constant extra space.

Example 1:

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

Example 2:

```
Input: nums = [0, 1, 0, 1, 0, 1, 99]  
Output: 99
```

Constraints:





except for one element which appears only once.

Seen this question in a real interview before?

Yes

No

[Subscribe](#) to see which companies asked this question.

[Related Topics ▾](#)

[Similar Questions ▾](#)

Py3



```
1 class Solution:
2     def singleNumber(self, nums:
List[int]) -> int:
3         if len(nums)==1:
4             return nums[0]
5         stack=[]
6         for i in nums:
7             if i in stack:
8                 stack.remove(i)
9             else:
10                stack.append(i)
11         return stack[0]
12
```





136. Single Number

Description

Hints

Submissions

Discuss

Pick One

Given a **non-empty** array of integers `nums`, every element appears twice except for one. Find that single one.

You must implement a solution with a linear runtime complexity and use only constant extra space.

Example 1:

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

Example 2:

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

Example 3:





question.

[Related Topics ▾](#)

[Similar Questions ▾](#)

Py3



```
1 class Solution:
2     def maxProfit(self, prices: List[int]) -> int:
3         minc=float("inf")
4         maxc=0
5         for k in prices:
6             if k<minc:
7                 minc=k
8             elif k-minc>maxc:
9                 maxc=k-minc
10            return maxc
```



121. Best Time to Buy and Sell Stock

Description

Hints

Submissions

Discuss

Pick One

You are given an array `prices` where `prices[i]` is the price of a given stock on the `ith` day.

You want to maximize your profit by choosing a **single day** to buy one stock and choosing a **different day in the future** to sell that stock.

Return *the maximum profit you can achieve from this transaction*. If you cannot achieve any profit, return

`0`.

Example 1:

Input: `prices = [7, 1, 5, 3, 6, 4]`

Output: `5`

Explanation: Buy on day 2 (`price = 1`)

Note that buying on day 2 and selling





Related Topics ▾

Py3



```
1 class Solution:
2     def minimumTotal(self,
triangle: List[List[int]]) ->
int:
3         h = triangle[-1][:]
4         k= len(triangle)
5
5         for i in range(k - 2, -1,
-1):
6             for j in
range(len(triangle[i])):
7                 h[j] =
triangle[i][j] + min(h[j],h[j +
1])
8
9         return h[0]
10
11
```



120. Triangle

Description

Hints

Submissions

Discuss

Pick One

Given a `triangle` array, return the *minimum path sum from top to bottom*.

For each step, you may move to an adjacent number of the row below. More formally, if you are on index `i` on the current row, you may move to either index `i` or index `i + 1` on the next row.

Example 1:

```
ngle = [[2], [3, 4], [6, 5, 7], [4, 1, 8, 3]]
```

: The triangle looks like:

```
path sum from top to bottom is 2 + 3 +
```

Example 2:





120. Triangle

Description

Hints

Submissions

Discuss

Pick One

Given a `triangle` array, return the *minimum path sum from top to bottom*.

For each step, you may move to an adjacent number of the row below. More formally, if you are on index `i` on the current row, you may move to either index `i` or index `i + 1` on the next row.

Example 1:

Input: triangle = [[2],[3,4],[6,5,7],[4,1,8,3]]

Output: 11

Explanation: The triangle looks like:

```
2  
3 4  
6 5 7  
4 1 8 3
```

The minimum path sum from top to bottom is 11.

Example 2:





Example 1:

Input: rowIndex = 3

Output: [1, 3, 3, 1]

Example 2:

Input: rowIndex = 0

Output: [1]

Example 3:

Input: rowIndex = 1

Output: [1, 1]

Constraints:

- `0 <= rowIndex <= 33`

Follow up: Could you optimize your algorithm to use only `O(rowIndex)` extra space?

Seen this question in a real interview before?



Yes

No





119. Pascal's Triangle II

Description

Hints

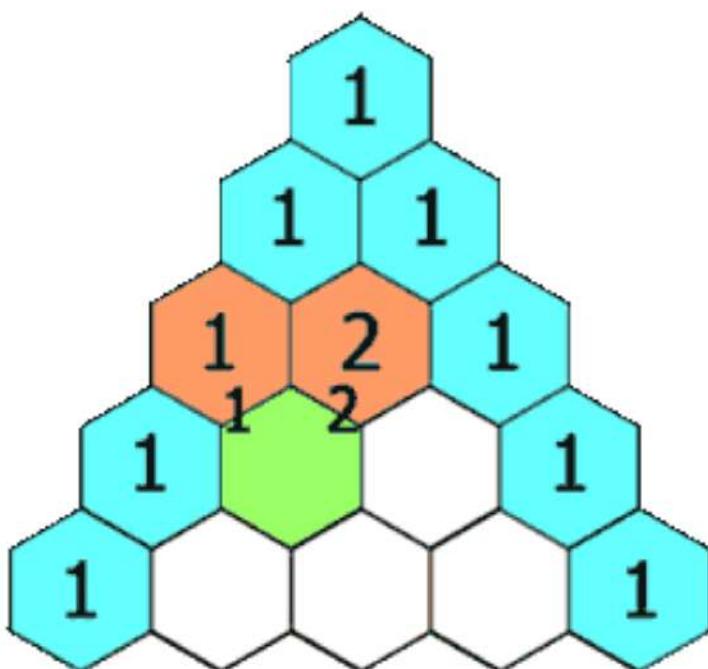
Submissions

Discuss

Pick One

Given an integer `rowIndex`, return the `rowIndexth` (0-indexed) row of the **Pascal's triangle**.

In **Pascal's triangle**, each number is the sum of the two numbers directly above it as shown:



[Related Topics ▾](#)[Similar Questions ▾](#)

Py3



```
1 class Solution:
2     import math
3     def generate(self, numRows:
4         int) -> List[List[int]]:
5         t=[]
6         for k in range(numRows):
7             r=[]
8             for j in range(k+1):
9                 r.append(math.comb(k,j))
10            t.append(r)
11        return t
12
13
```



118. Pascal's Triangle

Description

Hints

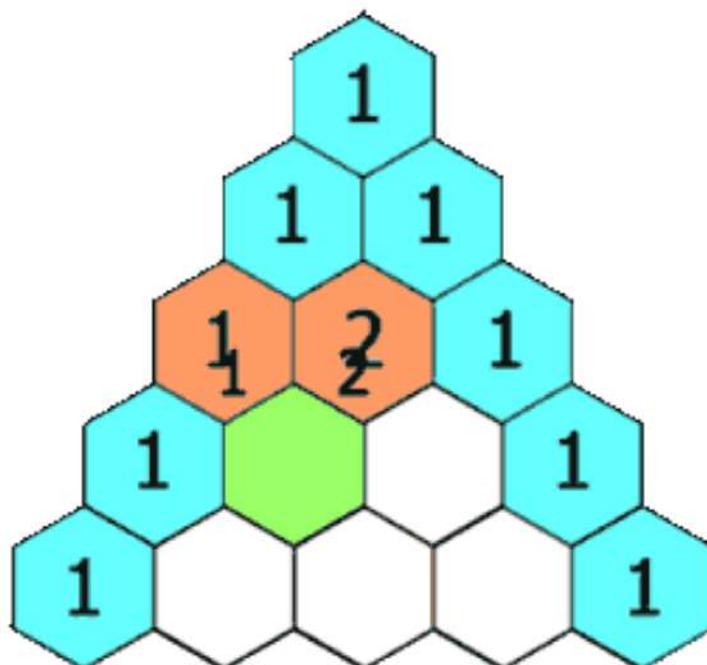
Submissions

Discuss

Pick One

Given an integer `numRows`, return the first `numRows` of **Pascal's triangle**.

In **Pascal's triangle**, each number is the sum of the two numbers directly above it as shown:



Example 1:



- `nums` is guaranteed to be rotated at some pivot.
- $-10^4 \leq \text{target} \leq 10^4$

Follow up: This problem is similar to [Search in Rotated Sorted Array](#), but `nums` may contain **duplicates**. Would this affect the runtime complexity? How and why?

Seen this question in a real interview before? 

Yes No

Subscribe to see which companies asked this question.

[Related Topics ▾](#)

[Similar Questions ▾](#)

Py3



```
1 class Solution:  
2     def search(self, nums:  
3         List[int], target: int) -> bool:  
4             for i in nums:  
5                 if i==target:  
6                     return True  
7             return False
```





Before being passed to your function, `nums` is **rotated** at an unknown pivot index `k` ($0 \leq k < \text{nums.length}$) such that the resulting array is `[nums[k], nums[k+1], ..., nums[n-1], nums[0], nums[1], ..., nums[k-1]]` (**0-indexed**). For example,

`[0,1,2,4,4,4,5,6,6,7]` might be rotated at pivot index `5` and become

`[4,5,6,6,7,0,1,2,4,4]`.

Given the array `nums` **after** the rotation and an integer `target`, return `true` if `target` is in `nums`, or `false` if it is not in `nums`.

You must decrease the overall operation steps as much as possible.

Example 1:

```
Input: nums = [2,5,6,0,0,1,2], target = 0
Output: true
```

Example 2:

```
Input: nums = [2,5,6,0,0,1,2], target
Output: false
```





Seen this question in a previous interview before?

Yes No

Subscribe to see which companies asked this question.

Related Topics ▾

Similar Questions ▾

Py3



```
1 class Solution:
2     def removeDuplicates(self,
3         nums: List[int]) -> int:
4             k=len(nums)
5             nums.sort()
6             if k<=2:
7                 return k
8             j=2
9             for i in range(2,k):
10                 if nums[i]!=nums[j-1]:
11                     nums[j]=nums[i]
12                     j=j+1
13
14             return j
```



If all assertions pass, then your solution will be accepted.

Example 1:

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

Output: 5, nums = [1,1,2,2,3,]

Explanation: Your function should return 5.
It does not matter what you leave beyond index 5.

Example 2:

Input: nums = [0,0,1,1,1,1,2,3,3]

Output: 7, nums = [0,0,1,1,2,3,3, ,]

Explanation: Your function should return 7.
It does not matter what you leave beyond index 7.

Constraints:

- `1 <= nums.length <= 3 * 104`
- `-104 <= nums[i] <= 104`
- `nums` is sorted in **non-decreasing** order.

Seen this question in a real interview before?



Yes

No





5445

1044



80. Remove Duplicates from Sorted Array II

Description

Hints

Submissions

Discuss

Pick One

Given an integer array `nums` sorted in **non-decreasing order**, remove some duplicates **in-place** such that each unique element appears **at most twice**. 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**





question.

[Related Topics ▾](#)

[Similar Questions ▾](#)

Py3



```
1 class Solution:
2     def plusOne(self, digits:
3         List[int]) -> List[int]:
4             k=int(''.join(map(str,
5             digits)))+1
6             return
7             list(map(int,str(k)))
```





66. Plus One

Description

Hints

Submissions

Discuss

Pick One

You are given a **large integer** represented as an integer array `digits`, where each `digits[i]` is the `ith` digit of the integer. The digits are ordered from most significant to least significant in left-to-right order. The large integer does not contain any leading `0`'s.

Increment the large integer by one and return *the resulting array of digits*.

Example 1:

```
: digits = [1,2,3]
t: [1,2,4]
nation: The array represents the integer 123. Incrementing by one gives 123 + 1 = 124.
the result should be [1,2,4].
```



- $1 \leq \text{nums.length} \leq 10^4$
- $0 \leq \text{nums[i]} \leq 10^5$

Seen this question in a real interview before?



Yes

No

Subscribe to see which companies asked this question.

Related Topics ▾

Similar Questions ▾

Py3



```
1 class Solution:  
2     def canJump(self, nums:  
3         List[int]) -> bool:  
4             n=len(nums)  
5             k=0  
6             for i in range (n):  
7                 if i>k:  
8                     return False  
9                 k=max(k,i+nums[i])  
return True
```





55. Jump Game

[Description](#)[Hints](#)[Submissions](#)[Discuss](#)[Pick One](#)

You are given an integer array `nums`. You are initially positioned at the array's **first index**, and each element in the array represents your maximum jump length at that position.

Return `true` if you can reach the last index, or `false` otherwise.

Example 1:

to 1, then 3 steps to the last index.

Example 2:

ys arrive at index 3 no matter what. It





55. Jump Game

Description

Hints

Submissions

Discuss

Pick One

You are given an integer array `nums`. You are initially positioned at the array's **first index**, and each element in the array represents your maximum jump length at that position.

Return `true` if you can reach the last index, or `false` otherwise.

Example 1:

```
From index 0 to 1, then 3 steps to the 1
```

Example 2:

```
Yes arrive at index 3 no matter what. It
```





55. Jump Game

Description
Hints
Submissions
Discuss
Pick One

You are given an integer array `nums`. You are initially positioned at the array's **first index**, and each element in the array represents your maximum jump length at that position.

Return `true` if you can reach the last index, or `false` otherwise.

Example 1:

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

Output: `true`

Explanation: Jump 1 step from index 0

Example 2:

Input: `nums = [3, 2, 1, 0, 4]`

Output: `false`

Explanation: You will always arrive at





Related Topics ▾

Similar Questions ▾

Py3



```
1 from itertools import
permutations
2 class Solution:
3     def permuteUnique(self, nums:
List[int]) -> List[List[int]]:
4         k=len(nums)
5         r=[]
6         for i in permutations
(nums,k):
7             r.append(i)
8
9         return set(r)
10
```