

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

You are standing at position `0` on an infinite number line. There is a goal at position `target`.

On each move, you can either go left or right. During the  $n$ -th move (starting from 1), you take  $n$  steps.

Return the minimum number of steps required to reach the destination.

#### Example 1:

**Input:** target = 3

**Output:** 2

**Explanation:**

On the first move we step from 0 to 1.

On the second step we step from 1 to 3.

#### Example 2:

**Input:** target = 2

**Output:** 3

**Explanation:**

On the first move we step from 0 to 1.

On the second move we step from 1 to -1.

On the third move we step from -1 to 2.

#### Note:

- `target` will be a non-zero integer in the range  $[-10^9, 10^9]$ .

Difficulty:

Easy

Total Accepted:

5.9K

Total Submissions:

19.9K

Contributor:

ngoc\_lam



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

[Related Topics ▾](#)

## 754. Reach a Number

[Description](#)[Hints](#)[Submissions](#)[Discuss](#)[Solution](#)[Nice Recursive Python Solution](#)[Share](#)

### Nice Recursive Python Solution

165 Views

Created at: August 20, 2018 6:53 AM

theodorewahle ★ 0

```
def reachNumber(self, target):
    def step(current_val, current_steps, target):
        next_step = current_steps + 1
        next_up = current_val + next_step
        next_down = current_val - next_step
        if next_up == target or next_down == target:
            return next_step
        elif next_up < target or next_down < target:
            return step(next_up, next_step, target)
        elif next_down > target or next_up > target:
            return step(next_down, next_step, target)
        current_val = 0
    return step(0, 0, target)
```

Comments: 1

Sort By ▾

## 66. Plus One

Description Hints Submissions Discuss Solution

Pick One

Given a **non-empty** array of digits representing a non-negative integer, plus one to the integer.

The digits are stored such that the most significant digit is at the head of the list, and each element in the array contain a single digit.

You may assume the integer does not contain any leading zero, except the number 0 itself.

**Example 1:**

```
Input: [1,2,3]
Output: [1,2,4]
Explanation: The array represents the integer 123.
```

**Example 2:**

```
Input: [4,3,2,1]
Output: [4,3,2,2]
Explanation: The array represents the integer 4321.
```

Difficulty: Easy

Total Accepted: 304.1K

Total Submissions: 760.1K

Contributor: LeetCode

[Share](#)

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

[Related Topics](#)

[Similar Questions](#)

Seen this question in a real interview before?

Python3 Array (4) Lowest Common Ancestor Plus One - LeetCode tree [C:\Users\saral... Run - tree Untitled - Notepad 114.flatten-binary-t... Questions and Ans... 9:56 PM

https://leetcode.com/problems/plus-one/discuss/185820/Python-In-place-solution-no-weird-one-liner-or-weird-functions-beats-99.7%

Python In place solution no weird one liner or weird functions, beats 99.7%

Created at: October 26, 2018 2:29 AM

oscargarza356 ★ 0 VIEWS

```
class Solution:
    def plusOne(self, digits):
        """
        :type digits: List[int]
        :rtype: List[int]
        """
        j = len(digits) - 1
        while j >= 0:
            if digits[j] == 9:
                digits[j] = 0
            if j == 0:
                digits.insert(0,1)
            else:
                digits[j] += 1
                break
            j -= 1
        return digits
```

Comments: 0

Sort By ▾

Windows Taskbar: Array (4) Lowest Common Ancestor Python In place sol... tree [C:\Users\saral... Run - tree Untitled - Notepad 114.flatten-binary-t... Questions and Ans... 9:56 PM

## 53. Maximum Subarray

Description Hints Submissions Discuss Solution

**Pick One**

Given an integer array `nums`, find the contiguous subarray (containing at least one number) which has the largest sum and return its sum.

**Example:**

```
Input: [-2,1,-3,4,-1,2,1,-5,4],  
Output: 6  
Explanation: [4,-1,2,1] has the largest sum = 6.
```

**Follow up:**  
If you have figured out the O(n) solution, try coding another solution using the divide and conquer approach, which is more subtle.

Seen this question in a real interview before?

Difficulty: **Easy**

Total Accepted: 391.1K

Total Submissions: 941K

Contributor: LeetCode

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

Related Topics ▾

Similar Questions ▾

Python3

## 53. Maximum Subarray

Description Hints Submissions **Discuss** Solution

Easy Python Way

5.3K VIEWS

Last Edit: October 25, 2018 12:40 AM

67

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

Comments: 29

Login to Comment

Arrays - InterviewBit × O(n) python - LeetCode × NDTV: Latest News, Inc. × Meri Mehbooba - Part 1 × Nice Recursive Python × LeetCode 114 Flatten Binary Tree × How to get a new page × +

https://leetcode.com/problems/maximum-subarray/discuss/184811/O(n)-python

Share

## O(n) python

161 VIEWS

Created at: October 23, 2018 1:18 PM

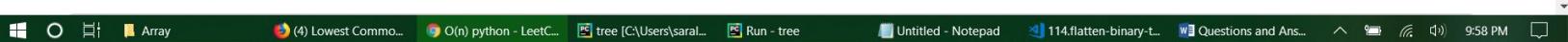
nothingleft123 ★ 0

0

```
class Solution:
    def maxSubArray(self, nums):
        """
        :type nums: List[int]
        :rtype: int
        """
        max_all = nums[0]
        max_current = 0
        for i in nums:
            if max_current + i < i:
                max_current = i
            else:
                max_current = max_current + i
            if max_all < max_current:
                max_all = max_current
        return max_all
    ...
```

Comments: 0

Sort By ▾



## Quick Links for Arrays

Recent Articles  
Practice Problems  
Quizzes

Videos

## Introduction

Introduction to Arrays

Arrays in C/C++

Arrays in Java

Arrays in Python

Explore more...

## Array Rotations

Program for array rotation

Reversal algorithm for array rotation

Block swap algorithm for array rotation

Program to cyclically rotate an array by one

Search an element in a sorted and rotated array

Given a sorted and rotated array, find if there is a pair with a given

## Maximum absolute difference of value and index sums

Given an unsorted array A of N integers,  $A_1, A_2, \dots, A_N$ . Return maximum value of  $f(i, j)$  for all  $1 \leq i, j \leq N$ .

$f(i, j)$  or absolute difference of two elements of an array A is defined as  $|A[i] - A[j]| + |i - j|$ , where  $|A|$  denotes the absolute value of A.

## Examples:

We will calculate the value of  $f(i, j)$  for each pair of  $(i, j)$  and return the maximum value thus obtained.

Input :  $A = \{1, 3, -1\}$ 

Output : 5

$$\begin{aligned} f(1, 1) &= f(2, 2) = f(3, 3) = 0 \\ f(1, 2) &= f(2, 1) = |1 - 3| + |1 - 2| = 3 \\ f(1, 3) &= f(3, 1) = |1 - (-1)| + |1 - 3| = 4 \\ f(2, 3) &= f(3, 2) = |3 - (-1)| + |2 - 3| = 5 \end{aligned}$$

So, we return 5.

Input :  $A = \{3, -2, 5, -4\}$ 

Output : 10

$$\begin{aligned} f(1, 1) &= f(2, 2) = f(3, 3) = f(4, 4) = 0 \\ f(1, 2) &= f(2, 1) = |3 - (-2)| + |1 - 2| = 6 \\ f(1, 3) &= f(3, 1) = |3 - 5| + |1 - 3| = 4 \\ f(1, 4) &= f(4, 1) = |3 - (-4)| + |1 - 4| = 10 \\ f(2, 3) &= f(3, 2) = |(-2) - 5| + |2 - 3| = 8 \\ f(2, 4) &= f(4, 2) = |(-2) - (-4)| + |2 - 4| = 4 \\ f(3, 4) &= f(4, 3) = |5 - (-4)| + |3 - 4| = 10 \end{aligned}$$

So, we return 10

## Trending Content

- Technical Scripter
- Geeks Classes
- Python List , Set , Tuple & Dictionary
- Number Theory
- Set to Array in Java
- BFS , DFS
- School Programming
- Longest Repeated Subsequence
- Longest Palindromic Subsequence
- Detect a negative cycle.
- GATE CS Notes

## Most Visited Posts

- Technical Scripter
- Top 10 algorithms in Interview Questions
- How to begin with Competitive Programming?
- Step by Step Guide for Placement Preparation
- How to prepare for ACM-ICPC?
- Insertion Sort , Binary Search , QuickSort , MergeSort , HeapSort

GeeksforGeeks

Rearrange an array such that  $arr[i] = i$ 

```
# Brute force Python 3 program
# to calculate the maximum
# absolute difference of an array.
```

Write a program to reverse an array or string

```
def calculateDiff(i, j, arr):
    # Utility function to calculate
    # the value of absolute difference
    # for the pair (i, j).
    return abs(arr[i] - arr[j]) + abs(i - j)
```

Rearrange array such that  $arr[i] \geq arr[j]$  if i is even and  $arr[i] \leq arr[j]$  if i is odd and  $j \leq i$ 

```
# Function to return maximum
# absolute difference in
# brute force.
```

Rearrange positive and negative numbers in  $O(n)$  time and  $O(1)$  extra space

```
# Variable for storing the
# maximum absolute distance
# throughout the traversal
# of loops.
```

Move all zeroes to end of array

```
result = 0
```

Move all zeroes to end of array | Set-2 (Using single traversal)

```
# Iterate through all pairs.
```

Minimum swaps required to bring all elements less than or equal to k together

```
for i in range(0,n):
    for j in range(i, n):
```

Rearrange positive and negative numbers using inbuilt sort function

```
# If the absolute difference of
# current pair (i, j) is greater
# than the maximum difference
# calculated till now, update
# the value of result.
```

Rearrange array such that even positioned are greater than odd

```
if (calculateDiff(i, j, arr) > result):
    result = calculateDiff(i, j, arr)
```

Explore more...

```
return result
```

## Order Statistics

K'th Smallest/Largest Element in Unsorted Array | Set 1 &amp; Expected Linear Time | Set 2 &amp; Worst Case Linear Time | Set 3

K'th Smallest/Largest Element using STL

```
# Driver program
arr = [-70, -64, -6, -56, 64,
       61, -57, 16, 48, -98]
n = len(arr)
```

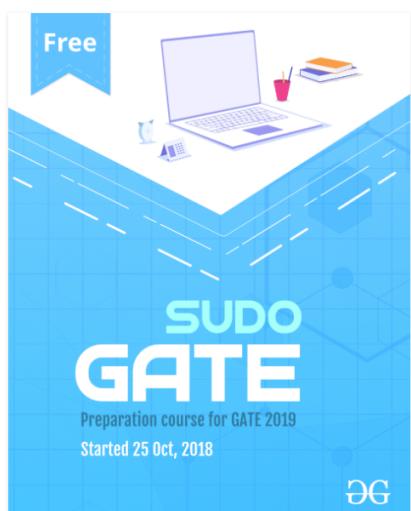
```
print(maxDistance(arr, n))
```

```
# This code is contributed by Smitha Dinesh Semwal
```

Run on IDE

Copy Code

A technical content writing event by GeeksforGeeks



## Tags

- Amazon Aptitude Aptitude Bit
- Magic C C C# C++ Computer Networks
- CPP-Functions C Quiz Dynamic Programming GBlog Geometric Graph Hash Internship
- Interview Experiences ISRO Java



class Solution:

```
def maxArr(self, A):
```

```
    lista=A
```

```
    Aminusl = lista[:]
```

```
    Aplusl = lista[:]
```

```
# print(Aminusl)
```

```
for i in range(len(lista)):
```

```
    # print(a[i],i+1)
```

```
    # print(a[i]- (i + 1))
```

```
Aminusl[i] = (Aminusl[i] - (i + 1))
```

```
Aplusl[i] = (Aplusl[i] + (i + 1))
```

```
# print(Aminusl)
```

```
# print(Aplusl)
```

```
apimax = max(Aplusl)
```

```
apimin = min(Aplusl)
```

```
amimax = max(Aminusl)
```

```
amimin = min(Aminusl)
```

```
valmax = max(apimax - apimin, amimax - amimin)
```

```
return(valmax)
```

Arrays - Interviewbit | Missing Number | Maximum absolute difference | NDTV: Latest News | Meri Mehbooba | Nice Recursive Python | LeetCode 114 Flatten | How to get a new | + | - | X | S

Description Hints Submissions Discuss Solution

Pick One

Given an array containing  $n$  distinct numbers taken from  $0, 1, 2, \dots, n$ , find the one that is missing from the array.

**Example 1:**

Input: [3,0,1]  
Output: 2

**Example 2:**

Input: [9,6,4,2,3,5,7,0,1]  
Output: 8

**Note:**  
Your algorithm should run in linear runtime complexity. Could you implement it using only constant extra space complexity?

Difficulty: Easy

Total Accepted: 214.5K

Total Submissions: 462.5K

Contributor: jianchao-li

f t G+ in

Subscribe to see which companies asked this question.

Related Topics ▾

Similar Questions ▾

Seen this question in a real interview before? Yes No

Python3

Arrays - Interviewbit | 3 line- Python 3 | Maximum absolute difference | NDTV: Latest News | Meri Mehbooba | Nice Recursive Python | LeetCode 114 Flatten | How to get a new | + | - | X | S

## 268. Missing Number

Description Hints Submissions Discuss Solution

3 line- Python 3 13 VIEWS

Created at: 19 hours ago 0 Share

0

```
class Solution:
    def missingNumber(self, nums):
        nums = set(nums)
        for i in range(len(nums)+1):
            if i not in nums: return i
```

Comments: 0 Sort By ▾

Login to Comment

```
def missingNumber(self, nums):
    """
    :type nums: List[int]
    :rtype: int
    """

    if not nums:
        return None

    total = sum(nums)
    n = len(nums)

    n_sum = (n * (n+1))//2

    return abs(n_sum-total)
```

☰ > Find k zeroes to be flipped so that number of consecutive 1's is maximized

Share

2.3K  
VIEWS

Last Edit: September 30, 2018 5:01 AM

bhanarkar ★3

3

Given a binary array and an integer k, find the position of zeroes flipping which creates maximum number of consecutive 1s in array.

Input: arr[] = {1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1}

m = 2

Output: 5 7

We are allowed to flip maximum 2 zeroes. If we flip arr[5] and arr[7], we get 8 consecutive 1's which is maximum possible under given constraints

Input: arr[] = {1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1}

m = 1

Output: 7

Input: arr[] = {0, 0, 0, 1}

m = 4

Output: 0 1 2

Comments: 6

Sort By ▾



Python : Using sliding window and queue.

```
def max_zeros(a, n):
    if not a:
        return 0

    from collections import deque
    queue = deque()
    count_zeros = 0
    max_len = 0

    i = 0
    while i < len(a):
        if a[i] == 1:
            queue.append(a[i])
            i += 1
        else:
            if count_zeros < n:
                queue.append(a[i])
                count_zeros += 1
                i += 1
            else:
                left = queue.popleft()
                if left == 0:
                    count_zeros -= 1
        max_len = max(max_len, len(queue))

    return max_len

max_zeros([1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1], 2)
```

## 287. Find the Duplicate Number

Description Hints Submissions Discuss Solution

**Pick One**

Given an array  $nums$  containing  $n + 1$  integers where each integer is between 1 and  $n$  (inclusive), prove that at least one duplicate number must exist. Assume that there is only one duplicate number, find the duplicate one.

**Example 1:**

```
Input: [1,3,4,2,2]
Output: 2
```

**Example 2:**

```
Input: [3,1,3,4,2]
Output: 3
```

**Note:**

1. You **must not** modify the array (assume the array is read only).
2. You must use only constant,  $O(1)$  extra space.
3. Your runtime complexity should be less than  $O(n^2)$ .
4. There is only one duplicate number in the array, but it could be repeated more than once.

Difficulty: Medium

Total Accepted: 139.4K

Total Submissions: 300.2K

Contributor: jianchao-li

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

[Related Topics](#) ▾

[Similar Questions](#) ▾

Arrays - Interviewbit (4) Missing Number... Find the Duplicate ... tree [C:\Users\saral... Run - tree Untitled - Notepad 114.flatten-binary-L... Questions and Ans... 10:10 PM

https://leetcode.com/problems/find-the-duplicate-number/discuss/185871/Simple-Python-Solution-in-O(n)-time-beats-99

287. Find the Duplicate Number

Description Hints Submissions **Discuss** Solution

Simple Python Solution in  $O(n)$  time, beats 99%

Share

51 VIEWS

Created at: October 26, 2018 5:01 AM

HaiderA12 ★ 0

```
class Solution:
    def findDuplicate(self, nums):
        t = {}
        for num in nums:
            if num in t:
                return num
            else:
                t[num] = 1
```

Comments: 1 Sort By ▾

10:11 PM

class Solution:

```
def repeatedNumber(self, A):
```

```
    a=list(A)
```

```
    for i in range (len(a)):
```

```
        ai=a[i]
```

```
        temp=a[i]
```

```
        temp=abs(temp)
```

```
        if (a[temp]<0):
```

```
            return(abs(a[i]))
```

```
        a[temp]=-a[temp]
```

## 179. Largest Number

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

Given a list of non negative integers, arrange them such that they form the largest number.

**Example 1:**

**Input:** [10, 2]  
**Output:** "210"

**Example 2:**

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

**Note:** The result may be very large, so you need to return a string instead of an integer.

Difficulty:

Medium

Total Accepted:

109.1K

Total Submissions:

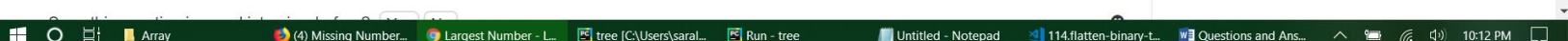
445.9K

Contributor:

ts



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

[Related Topics ▾](#)


```
# build-in function
def largestNumber1(self, nums):
    if not any(nums):
        return "0"
    return "".join(sorted(map(str, nums), cmp=lambda n1, n2: -1 if n1+n2>n2+n1 else (1 if n1+n2<n2+n1 else 0)))

# bubble sort
def largestNumber2(self, nums):
    for i in xrange(len(nums), 0, -1):
        for j in xrange(i-1):
            if not self.compare(nums[j], nums[j+1]):
                nums[j], nums[j+1] = nums[j+1], nums[j]
    return str(int("".join(map(str, nums))))

def compare(self, n1, n2):
    return str(n1) + str(n2) > str(n2) + str(n1)

# selection sort
def largestNumber3(self, nums):
    for i in xrange(len(nums), 0, -1):
        tmp = 0
        for j in xrange(i):
            if not self.compare(nums[j], nums[tmp]):
```

```

tmp = j
nums[tmp], nums[i-1] = nums[i-1], nums[tmp]
return str(int("".join(map(str, nums)))))

# insertion sort

def largestNumber4(self, nums):
    for i in xrange(len(nums)):
        pos, cur = i, nums[i]
        while pos > 0 and not self.compare(nums[pos-1], cur):
            nums[pos] = nums[pos-1] # move one-step forward
            pos -= 1
        nums[pos] = cur
    return str(int("".join(map(str, nums))))

# merge sort

def largestNumber5(self, nums):
    nums = self.mergeSort(nums, 0, len(nums)-1)
    return str(int("".join(map(str, nums))))

def mergeSort(self, nums, l, r):
    if l > r:
        return
    if l == r:
        return [nums[l]]
    mid = l + (r-l)//2
    left = self.mergeSort(nums, l, mid)
    right = self.mergeSort(nums, mid+1, r)
    return self.merge(left, right)

def merge(self, l1, l2):
    res, i, j = [], 0, 0
    while i < len(l1) and j < len(l2):
        if not self.compare(l1[i], l2[j]):
            res.append(l2[j])
            j += 1
        else:
            res.append(l1[i])
            i += 1
    res.extend(l1[i:] or l2[j:])
    return res

```

```
# quick sort, in-place

def largestNumber(self, nums):
    self.quickSort(nums, 0, len(nums)-1)
    return str(int("".join(map(str, nums)))))

def quickSort(self, nums, l, r):
    if l >= r:
        return
    pos = self.partition(nums, l, r)
    self.quickSort(nums, l, pos-1)
    self.quickSort(nums, pos+1, r)

def partition(self, nums, l, r):
    low = l
    while l < r:
        if self.compare(nums[l], nums[r]):
            nums[l], nums[low] = nums[low], nums[l]
            low += 1
        l += 1
    nums[low], nums[r] = nums[r], nums[low]
    return low
```

**Technical Scripter**

**Quick Links for Arrays**

- Recent Articles
- Practice Problems
- Quizzes
- Videos

**Introduction**

- Introduction to Arrays
- Arrays in C/C++
- Arrays in Java
- Arrays in Python
- Explore more...

**Array Rotations**

- Program for array rotation
- Reversal algorithm for array rotation
- Block swap algorithm for array

## Given an array arr[], find the maximum j – i such that arr[j] > arr[i]

Given an array arr[], find the maximum j – i such that arr[j] > arr[i].

### Examples :

Input: {34, 8, 10, 3, 2, 80, 30, 33, 1}  
Output: 6 (j = 7, i = 1)

Input: {9, 2, 3, 4, 5, 6, 7, 8, 18, 0}  
Output: 8 (j = 8, i = 0)

Input: {1, 2, 3, 4, 5, 6}  
Output: 5 (j = 5, i = 0)

Input: {6, 5, 4, 3, 2, 1}  
Output: -1

**Trending Content**

- Technical Scripter
- Geeks Classes
- Python List , Set , Tuple & Dictionary
- Number Theory
- Set to Array in Java
- BFS , DFS
- School Programming
- Longest Repeated Subsequence
- Longest Palindromic Subsequence
- Detect a negative cycle.
- GATE CS Notes

**Recommended:** Please solve it on “[PRACTICE](#)” first, before moving on to the solution.

Max Distance - InterviewBit Given an array arr[] find the max https://www.geeksforgeeks.org/given-an-array-arr-find-the-maximum-j-i-such-that-arrj-arr/ Given an array... array2 [C:\Use... Untitled - Not... Result - VS Py... Jupyter Noteb python - Com... Python 3.6 (64... Questions an... 10:10 PM

Search an element in a sorted and rotated array

Given a sorted and rotated array, find if there is a pair with a given sum

Find maximum value of Sum(i\*arr[i]) with only rotations on given array allowed

Maximum sum of i\*arr[i] among all rotations of a given array

Find the Rotation Count in Rotated Sorted array

Quickly find multiple left rotations of an array

Explore more...

**Arrangement Rearrangement**

Rearrange an array such that arr[i] = i

Write a program to reverse an array or string

Rearrange array such that arr[i] ≥ arr[i] if i is even and arr[i] ≤ arr[i] if i is odd and j ≤ i

Rearrange positive and negative numbers in O(n) time and O(1) extra space

**Python3**

```
# Python3 program to find the maximum
# j - i such that arr[j] > arr[i]

# For a given array arr[], returns
# the maximum j - i such that
# arr[j] > arr[i]
def maxIndexDiff(arr, n):
    maxDiff = -1
    for i in range(0, n):
        j = n - 1
        while(j > i):
            if arr[j] > arr[i] and maxDiff < (j - i):
                maxDiff = j - i;
            j -= 1

    return maxDiff

# driver code
arr = [9, 2, 3, 4, 5, 6, 7, 8, 18, 0]
n = len(arr)
maxDiff = maxIndexDiff(arr, n)
print(maxDiff)

# This article is contributed by Smitha Dinesh Semwal
```

Run on IDE Copy Code

**GeeksforGeeks**  
A computer science portal for geeks

**TECHNICAL SCRIPTER**

**2 OCT to 2 DEC**

A technical content writing event by GeeksforGeeks

**Free**

Articles > 624. Maximum Distance in Array



### Approach #1 Brute Force [Time Limit Exceeded]

The simplest solution is to pick up every element of every array from the *list* and find its distance from every element in all the other arrays except itself and find the largest distance from out of those.

```
Java Copy
1 public class Solution {
2     public int maxDistance(int[][] list) {
3         int res = 0;
4         for (int i = 0; i < list.length - 1; i++) {
5             for (int j = 0; j < list[i].length; j++) {
6                 for (int k = i + 1; k < list.length; k++) {
7                     for (int l = 0; l < list[k].length; l++) {
8                         res = Math.max(res, Math.abs(list[i][j] - list[k][l]));
9                     }
10                }
11            }
12        }
13        return res;
14    }
15 }
16 }
```

## Complexity Analysis

- Time complexity :  $O((n * x)^2)$ . We traverse over all the arrays in *list* for every element of every array considered. Here,  $n$  refers to the number of arrays in the *list* and  $x$  refers to the average number of elements in each array in the *list*.

https://www.geeksforgeeks.org/given-an-array-arr-find-the-maximum-j-i-such-that-arr[j]>arr[i]

OG Algo DS Languages Interview Students GATE CS Subjects Quizzes GBlog Puzzles Courses

rotation

Block swap algorithm for array rotation

Program to cyclically rotate an array by one

Search an element in a sorted and rotated array

Given a sorted and rotated array, find if there is a pair with a given sum

Find maximum value of Sum( $i \cdot arr[i]$ ) with only rotations on given array allowed

Maximum sum of  $i \cdot arr[i]$  among all rotations of a given array

Find the Rotation Count in Rotated Sorted array

Quickly find multiple left rotations of an array

Explore more...

**Arrangement Rearrangement**

Rearrange an array such that  $arr[i] = i$

Write a program to reverse an array or string

Rearrange array such that  $arr[i] >$

**Method 1 (Simple but Inefficient)**

Run two loops. In the outer loop, pick elements one by one from left. In the inner loop, compare the picked element with the elements starting from right side. Stop the inner loop when you see an element greater than the picked element and keep updating the maximum  $j - i$  so far.

C++ C Java Python3 C# PHP

```
# Python3 program to find the maximum
# j - i such that arr[j] > arr[i]

# For a given array arr[], returns
# the maximum j - i such that
# arr[j] > arr[i]
def maxIndexDiff(arr, n):
    maxDiff = -1
    for i in range(0, n):
        j = n - 1
        while(j > i):
            if arr[j] > arr[i] and maxDiff < (j - i):
                maxDiff = j - i;
            j -= 1

    return maxDiff

# driver code
arr = [9, 2, 3, 4, 5, 6, 7, 8, 18, 0]
n = len(arr)
maxDiff = maxIndexDiff(arr, n)
print(maxDiff)

# This article is contributed by Smitha Dinesh Semwal
```

Run on IDE Copy Code

Step by Step Guide for Placement Preparation

How to prepare for ACM-ICPC?

Insertion Sort , Binary Search , QuickSort , MergeSort , HeapSort

GeeksforGeeks  
A computer science portal for geeks

TECHNICAL SCRIPTER

2 OCT to 2 DEC

A technical content writing event by GeeksforGeeks

Free

## Technical Scripter

## Quick Links for Arrays

Recent Articles

Practice Problems

Quizzes

Videos

## Introduction

Introduction to Arrays

Arrays in C/C++

Arrays in Java

Arrays in Python

Explore more...

## Array Rotations

Program for array rotation

## Given an array arr[], find the maximum j – i such that arr[j] > arr[i]

Given an array arr[], find the maximum j – i such that arr[j] > arr[i].

**Examples :**

Input: {34, 8, 10, 3, 2, 80, 30, 33, 1}  
Output: 6 (j = 7, i = 1)

Input: {9, 2, 3, 4, 5, 6, 7, 8, 18, 0}  
Output: 8 (j = 8, i = 0)

Input: {1, 2, 3, 4, 5, 6}  
Output: 5 (j = 5, i = 0)

Input: {6, 5, 4, 3, 2, 1}  
Output: -1

**Recommended:** Please solve it on “[PRACTICE](#)” first, before moving on to the solution.

## Trending Content

Technical Scripter

Geeks Classes

Python List , Set , Tuple &amp; Dictionary

Number Theory

Set to Array in Java

BFS , DFS

School Programming

Longest Repeated Subsequence

Longest Palindromic Subsequence

Detect a negative cycle.

GATE CS Notes

## Most Visited Posts

Technical Scripter

Top 10 algorithms in Interview Questions

simple Python O(n) time, O(1) space solution

Created at: October 23, 2018 2:14 AM

0

131 VIEWS

The idea is to encode that we've seen a number `n` by making `nums[n]` negative. There is a duplicate when you look at another element in `num` whose value is `n` because `nums[n]` is negative.

```
class Solution:
    def findDuplicate(self, nums):
        """
        :type nums: List[int]
        :rtype: int
        """
        for i in range(len(nums)):
            n = abs(nums[i])
            if nums[n] < 0:
                return n
            else:
                nums[n] *= -1
        return None
```

## 59. Spiral Matrix II

[Description](#) [Hints](#) [Submissions](#) [Discuss](#) [Solution](#)

**Pick One**

Given a positive integer  $n$ , generate a square matrix filled with elements from 1 to  $n^2$  in spiral order.

**Example:**

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

Difficulty: Medium

Total Accepted: 116.2K

Total Submissions: 266.6K

Contributor: LeetCode

[f](#) [t](#) [G+](#) [in](#)

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

[Related Topics](#)

[Similar Questions](#)

Seen this question in a real interview before?

Python3

[Copy](#) [Run](#) [Share](#) [Close](#)

```
1 class Solution:
    pass
```

20ms python solution beats 100%

```
0
def generateMatrix(self, n):
    """
    :type n: int
    :rtype: List[List[int]]
    """

    matrix = [[0] * n for _ in range(n)]
    l, r, u, d = 0, n-1, 0, n-1
    num = 1

    while l < r and u < d:
        for i in range(l, r):
            matrix[l][i] = num
            num += 1

        for j in range(u, d):
            matrix[j][d] = num
            num += 1

        for z in range(r, l, -1):
            matrix[r][z] = num
            num += 1

        for h in range(d, u, -1):
            matrix[h][u] = num
            num += 1

        l += 1
        r -= 1
        u += 1
        d -= 1

    if l == r or u == d:
        matrix[l][r] = num

    return matrix
```

```
class Solution(object):

    def generateMatrix(self, n):
        """
        :type n: int
        :rtype: List[List[int]]
        """

        if n==0 : return ([[ ]])
        if n==1 : return ([[1]])
        tmp, ctr, matrix = 1, 0, [[0 for j in range(n)] for i in range(n)]
        while tmp < n*n:
            for j in range(ctr, n-ctr-1): matrix[ctr][j] = tmp; tmp+=1
            for i in range(ctr, n-ctr-1): matrix[i][-ctr+1] = tmp; tmp+=1
            for j in range(ctr+1, n-ctr)[::-1]: matrix[-ctr][j] = tmp; tmp+=1
            for i in range(ctr+1, n-ctr)[::-1]: matrix[i][ctr] = tmp; tmp+=1
            ctr += 1
        if n*n-tmp==0:
            matrix[ctr][ctr]=n*n
        return matrix
```

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

Difficulty:

Easy

Total Accepted: 201.4K

Total Submissions: 473.1K

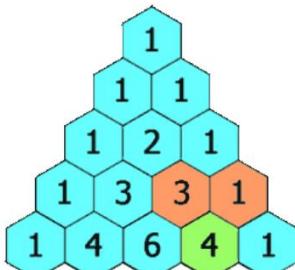
Contributor: LeetCode



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

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

Given a non-negative integer `numRows`, generate the first `numRows` of Pascal's triangle.



In Pascal's triangle, each number is the sum of the two numbers directly above it.

**Example:**

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

Python code 20ms

Share

VIEWS

Created at: 3 days ago

andizzlezhang ★ 5

0

▼

```
class Solution(object):
    def generate(self, numRows):
        """
        :type numRows: int
        :rtype: List[List[int]]
        """

        if numRows == 0:
            return []
        ans = [[1], [1, 1]]
        if numRows < 3:
            return ans[:numRows]

        numRows -= 2
        while numRows:
            prev = ans[-1]
            ans.append([1] + [prev[i] + prev[i+1] for i in range(0, len(prev) - 1)] + [1])
            numRows -= 1
        return ans
```

Comments: 0

Sort By ▾

## Elegant python solution

The point is to notice that from n to n+1, one row (call it R) need to be added. Let r be the last element for pascal-n, then:

```
tmp1 = [0] + r; tmp2 = r + [0],  
R = [a+b for a, b in zip(tmp1, tmp2)].
```

```
class Solution:  
    def generate(self, numRows):  
        """  
        :type numRows: int  
        :rtype: List[List[int]]  
        """  
        n = numRows  
        if n == 0:  
            return []  
  
        if n == 1:  
            return [[1]]  
  
        pascal = [[1]]  
        for i in range(2, n+1):  
            tmp1 = [0] + pascal[-1]  
            tmp2 = pascal[-1] + [0]  
            tmp = [a+b for a, b in zip(tmp1, tmp2)]  
            pascal.append(tmp)  
        return pascal
```

Comments: 0

Sort By ▾

```
def generate(numRows):  
    pascal = [[1]*(i+1) for i in range(numRows)]  
    for i in range(numRows):  
        for j in range(1,i):  
            pascal[i][j] = pascal[i-1][j-1] + pascal[i-1][j]  
    return pascal
```

## 31. Next Permutation

Description Hints Submissions Discuss Solution

**Pick One**

Difficulty: Medium

Total Accepted: 184.2K

Total Submissions: 627.7K

Contributor: LeetCode

f t G+ in

Subscribe to see which companies asked this question.

Related Topics ▾

Similar Questions ▾

Implement **next permutation**, which rearranges numbers into the lexicographically next greater permutation of numbers. If such arrangement is not possible, it must rearrange it as the lowest possible order (ie, sorted in ascending order). The replacement must be **in-place** and use only constant extra memory.

Here are some examples. Inputs are in the left-hand column and its corresponding outputs are in the right-hand column.

1,2,3	→ 1,3,2
3,2,1	→ 1,2,3
1,1,5	→ 1,5,1

Seen this question in a real interview before? Yes No

Arrays - Interviewbit X Python recursive sol... X Find k zeroes to be X NDTV: Latest News X Meri Mehbooba X Nice Recursive Pyt... X LeetCode 114 Flat! X How to get a new X 10:32 PM

Python recursive solution

Finally get [4,2,0,3,0,2,2]

```
class Solution:
    def nextPermutation(self, nums):
        """
        :type nums: List[int]
        :rtype: void Do not return anything, modify nums in-place instead.
        """
        if nums is None or len(nums) < 2:
            return
        end = len(nums)-1
        def permutation_iter(nums, front=0):
            for i in range(end-1, front-1, -1):
                for j in range(end, i, -1):
                    # swap
                    if nums[j] > nums[i]:
                        nums[j], nums[i] = nums[i], nums[j]
                        permutation_iter(nums, front=i+1)
                        return
            # if it is a descending list, reverse
            step = int((len(nums)-front)/2)
            for i in range(step):
                nums[front+i], nums[end-i] = nums[end-i], nums[front+i]
        permutation_iter(nums)
```

Comments: 0 Sort By ▾

Loop from end and find `target` where `nums[target+1] > nums[target]`. we then reverse the elements after `target` so that it's sorted in ascending order. we then swap `nums[target]` with `nums[j]` where number at `j` is the first number bigger than `nums[target]` which is always the smallest number that's bigger than `nums[target]` since we now have them sorted.

example: `[1,2,3,2,7,5,1]` (answer: `[1,2,3,5,1,2,7]`). `target` would be index 3. We sort (reverse) the number after `target` which gives us `[1,2,3,2,1,5,7]`. We would then find the number at `j` which is bigger than `nums[target]` and swap the two giving us `[1,2,3,5,1,2,7]`

```
def nextPermutation(self, nums):
    """
    :type nums: List[int]
    :rtype: void Do not return anything, modify nums in-place instead.
    """
    target=0
    for i in range(len(nums)-1, -1, -1):
        if i == 0:
            nums.reverse()
            return
        if nums[i] > nums[i-1]:
            target = i-1
            break

    i, j = target+1, len(nums)-1

    while(i < j):
        nums[i], nums[j] = nums[j], nums[i]
        i+=1
        j-=1

    for i in range(target+1, len(nums)):
        if nums[i] > nums[target]:
            nums[i], nums[target] = nums[target], nums[i]
            break
```