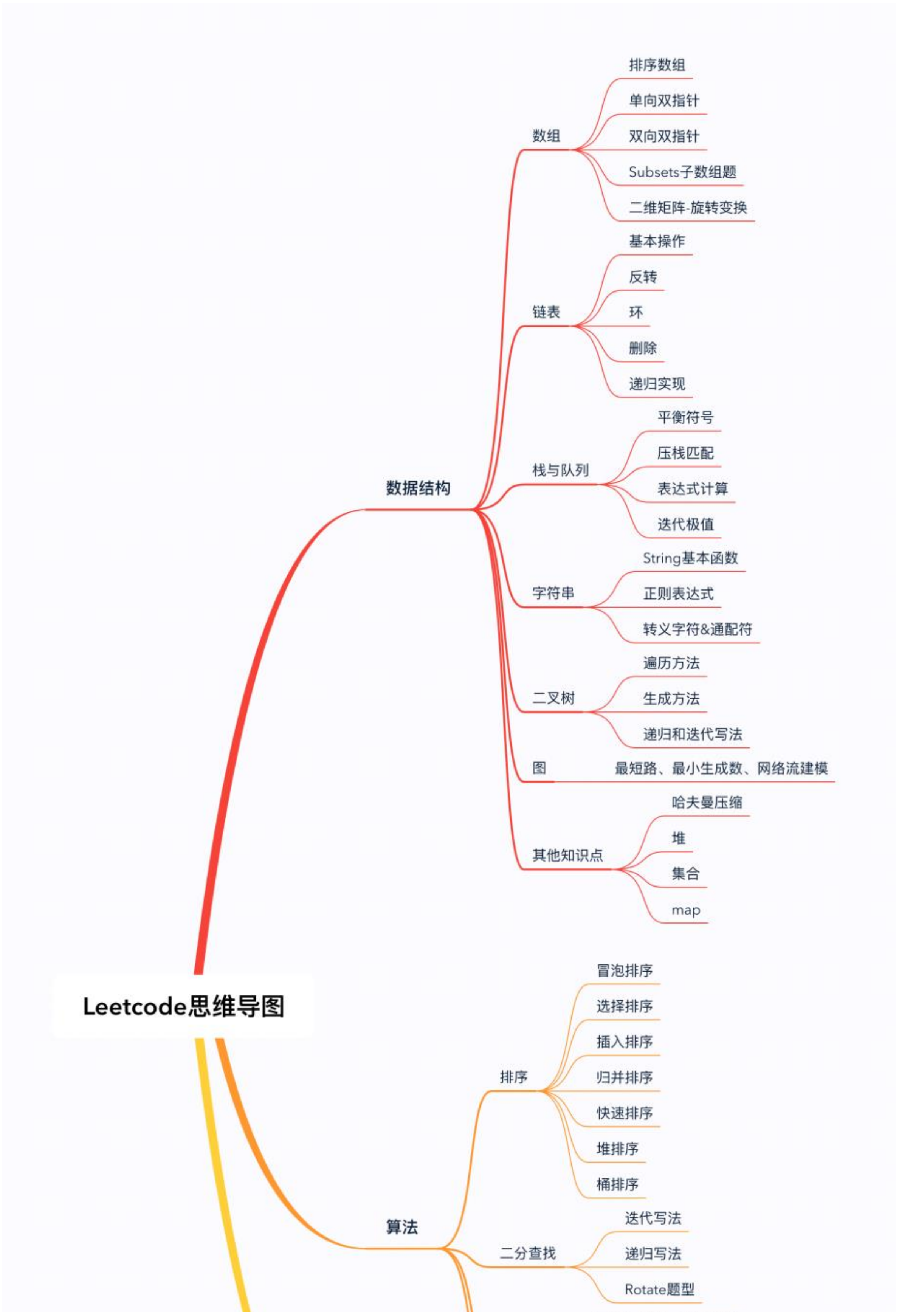
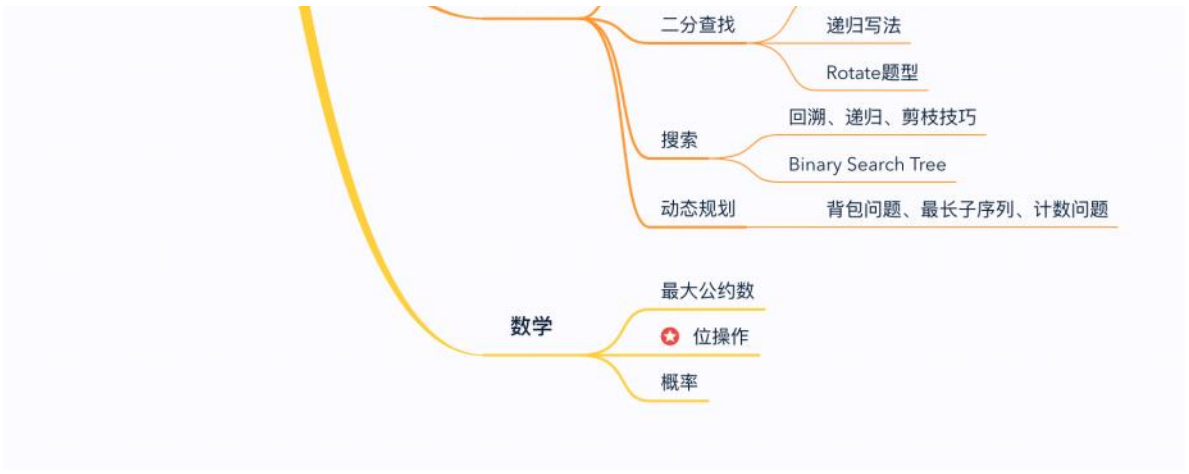
Wednesday, June 12, 2019 3:08 PM




二分查找
递归写法
Rotate题型
搜索
回溯、递归、剪枝技巧
Binary Search Tree
动态规划
背包问题、最长子序列、计数问题
数学
最大公约数
位操作
概率

From <https://www.cnblogs.com/charlotte77/p/10409417.html>

Wednesday, June 12, 2019 5:14 PM

Leetcode List

26-remove-duplicate...

*会做但有别的方法

#边缘问题出错

%不会做

什么都没标，会做且没什么更好的方法

189-rotate-array

27-remove-element

------------Easy 109-------

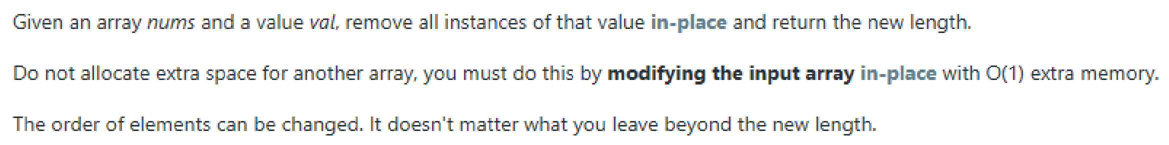
Sunday, June 16, 2019 9:50 PM

*27. Remove Element

Wednesday, June 12, 2019 3:31 PM

Description

Link: https://leetcode.com/problems/remove-element/



Given an array *nums* and a value *val*, remove all instances of that value **in-place** and return the new length.

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

The order of elements can be changed. It doesn't matter what you leave beyond the new length.

注意这里remove的定义是取nums的前length个元素

Tag: Array, Two pointers

Solution

- Approach1: Two pointers

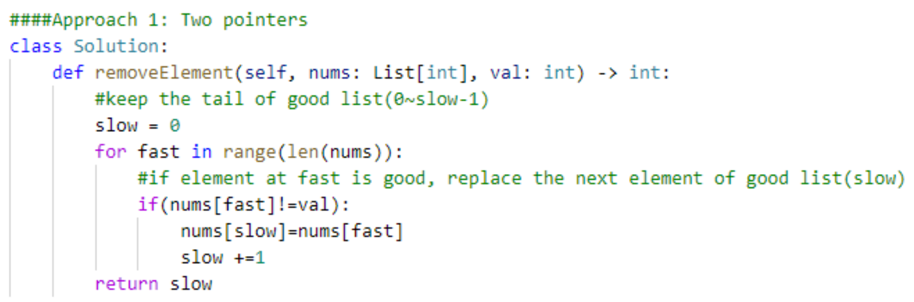
Intuition
slow: keep the tail of good list(0~slow-1)
if element at fast is good, replace the next element of good list(slow)

Complexity

Time: O(n)
Space: O(1)

Key: Two pointers

Code



```
####Approach 1: Two pointers
class Solution:
    def removeElement(self, nums: List[int], val: int) -> int:
        #keep the tail of good list(0~slow-1)
        slow = 0
        for fast in range(len(nums)):
            #if element at fast is good, replace the next element of good list(slow)
            if(nums[fast]!=val):
                nums[slow]=nums[fast]
                slow +=1
        return slow
```

- Approach2: Two pointers- If remove elements are rare

Intuition
if meet bad element, replace it by #n-1
nums=[4,1,2,3,5],val=4. It seems unnecessary to move elements [1,2,3,5]

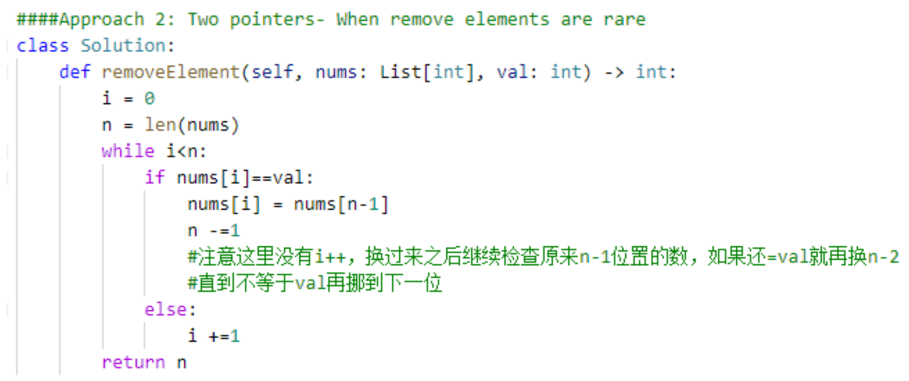
Complexity

Time: O(n)
Space: O(1)

Key: Two pointers

symmetric to Approach 1

Code



```
####Approach 2: Two pointers- When remove elements are rare
class Solution:
    def removeElement(self, nums: List[int], val: int) -> int:
        i = 0
        n = len(nums)
        while i<n:
            if nums[i]==val:
                nums[i] = nums[n-1]
                n -=1
                #注意这里没有i++，换过来之后继续检查原来n-1位置的数，如果还=val就再换n-2
                #直到不等于val再挪到下一位
            else:
                i +=1
        return n
```

#26.Remove Duplicates from Sorted Array

Wednesday, June 12, 2019 6:33 PM

- **Description**

  Link: https://leetcode.com/problems/remove-duplicates-from-sorted-array/

  Given a sorted array *nums*, remove the duplicates **in-place** such that each element appear only *once* and return the new length.

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

- **Tag: Array**
- **Solution**

• Approach1:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key: 注意list长度为0的情况

```
#### Approach 1: Two pointers

class Solution:
    def removeDuplicates(self, nums: List[int]) -> int:
        if len(nums)==0: return 0; #注意数据长度为0的特例
        slow = 1
        cur_val = nums[0]
        for elem in nums:
            if(elem!=cur_val):
                nums[slow]=elem
                cur_val = elem
                slow +=1
        return slow
```

• Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

•

#189.Rotate Array

Wednesday, June 12, 2019 8:15 PM

- **Description**

  Link: https://leetcode.com/problems/rotate-array

  Given an array, rotate the array to the right by *k* steps, where *k* is non-negative.

  Note:

  - Try to come up as many solutions as you can, there are at least 3 different ways to solve this problem.
  - Could you do it in-place with O(1) extra space?

- **Tag: , Array**
- **Solution**

- Approach1: Cyclic replacement

Intuition

Your input

```
[1,2,3,4,5,6,7]
3
```

stdout

```
[1, 2, 3, 1, 5, 6, 7]
[1, 2, 3, 1, 5, 6, 4]
[1, 2, 7, 1, 5, 6, 4]
[1, 2, 7, 1, 5, 3, 4]
[1, 6, 7, 1, 5, 3, 4]
[1, 6, 7, 1, 2, 3, 4]
[5, 6, 7, 1, 2, 3, 4]
```

Complexity

Time: O(n)
Space: O(1)

Code

```python
class Solution(object):
    def rotate(self, nums, k):
        """
        :type nums: List[int]
        :type k: int
        :rtype: None Do not return anything, modify nums in-place instead.
        """
        k = k % len(nums)
        count = 0
        start = 0
        while count < len(nums):
            current = start
            prev = nums[start] #store the value in the position

            while True:
                next = (current + k) % len(nums) #
                temp = nums[next] #store it temporaly
                nums[next] = prev #overide the next
                prev = temp #reset prev
                current = next #reset current
                count += 1

                if start == current:
                    break
            start += 1
```

Key: how to move element to right by k:

```python
prev = nums[start] #store the value in the position

while True:
    next = (current + k) % len(nums) #
    temp = nums[next] #store it temporaly
    nums[next] = prev #overide the next
    prev = temp #reset prev
    current = next #reset current
```

- ## Approach2: Reverse Array

Intuition

Let $n = 7$ and $k = 3$.

```
Original List                   : 1 2 3 4 5 6 7
After reversing all numbers     : 7 6 5 4 3 2 1
After reversing first k numbers : 5 6 7 4 3 2 1
After revering last n-k numbers : 5 6 7 1 2 3 4 --> Result
```

Complexity

Time: O(n)
Space: O(1)

Code

```
def rotate(self, nums, k):
    """
    :type nums: List[int]
    :type k: int
    :rtype: None Do not return anything, modify nums in-place instead.
    """
    k %= len(nums)
    self.reverse(nums,0,len(nums)-1)
    self.reverse(nums,0, k-1)
    self.reverse(nums,k, len(nums)-1)

def reverse(self, nums, start, end):
    """
    :type nums: List[int]
    :type start: int
    :type end: int
    :rtype: None
    """
    while start < end: #
        temp = nums[start]
        nums[start] = nums[end]
        nums[end] = temp
        start += 1
        end -= 1
```

Key:

- how to reverse array:

```
while start < end: #
    temp = nums[start]
    nums[start] = nums[end]
    nums[end] = temp
    start += 1
    end -= 1
```

- Self.function() to use external function

- ## My approach: Extra Array

Intuition

Save the last k element, move the rest element to right by k

Complexity

Time: O(k+n)
Space: O(k)

Code

Key: Consider if k>length

```
def rotate(self, nums, k):
    """
    :type nums: List[int]
    :type k: int
    :rtype: None Do not return anything, modify nums in-place instead.
    """
    length = len(nums)
    saveK=[]
    step = length-(k%length)# deal with the case that k>length
    for i in range(length+step):
        if i<step:
            saveK.append(nums[i])
        elif i<length:
            nums[i-step]=nums[i]
        else:
            nums[i-step]=saveK[i-length]
```

118. Pascal's Triangle

Sunday, June 16, 2019 7:53 PM

Description

Link: https://leetcode.com/problems/pascals-triangle/

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

Tag: Array

Solution

- My Approach:

Intuition

Complexity

Time: O(n^2)
Space: O(n^2)

Code

Key:

```
class Solution(object):
    def generate(self, numRows):
        """
        :type numRows: int
        :rtype: List[List[int]]
        """
        List = []
        for i in range(numRows):
            row =[]
            row.append(1)
            if i==0:
                List.append(row)
                continue
            elif i==1:
                row.append(1)
                List.append(row)
                continue
            else:
                for j in range(len(List[i-1])-1):
                    row.append(List[i-1][j]+List[i-1][j+1])
            row.append(1)
            List.append(row)
        return List
```

119. Pascal's Triangle II

Sunday, June 16, 2019 8:21 PM

- **Description**

  Link: https://leetcode.com/problems/pascals-triangle-ii/

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

- **Tag: Array**
- **Solution**

- My approach:

Intuition

Complexity

Time: O(n^2)
Space: O(2*n)

Key:

Code

```
class Solution(object):
    def getRow(self, rowIndex):
        """
        :type rowIndex: int
        :rtype: List[int]
        """
        for i in range(rowIndex+1):
            row =[]
            row.append(1)
            if i==0:
                lastrow = row
                continue
            elif i==1:
                row.append(1)
                lastrow = row
                continue
            else:
                for j in range(len(lastrow)-1):
                    row.append(lastrow[j]+lastrow[j+1])
            row.append(1)
            lastrow = row
        return row
```

*169. Majority Element

Sunday, June 16, 2019 8:40 PM

Description

Link: https://leetcode.com/problems/majority-element/

Given an array of size n, find the majority element. The majority element is the element that appears more than ⌊ n/2 ⌋ times.
You may assume that the array is non-empty and the majority element always exist in the array.

Tag: Array

Solution

- Approach1: Sort

Intuition
Since the majority value is larger than n/2, so the element in the middle is the element we want no matter which side have same values.

Complexity

Time: O(nlogn)
Space: O(1) or O(n)
Depends on if it is in-place
Sort function in Java& Python is O(n)

Code

```
class Solution:
    def majorityElement(self, nums):
        nums.sort()
        return nums[len(nums)//2]
```

Key: we can use sort function

- Approach2: Boyer-Moore Voting Algorithm

Intuition
When count != 0 , it means nums[1...i] has a majority,which is major in the solution.
When count == 0 , it means nums[1...i] doesn't have a majority, so nums[1...i] will not help nums[1...n].And then we have a subproblem of nums[i+1...n].

Complexity

Time: O(n)
Space: O(1)

Code

Key: Boyer-Moore Voting Algorithm

```
def majorityElement(self, nums):
    count = 0
    candidate = None

    for num in nums:
        if count == 0:
            candidate = num
        count += (1 if num == candidate else -1)

    return candidate
```

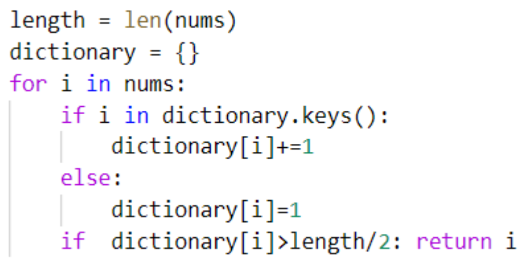
Sunday, June 16, 2019 9:44 PM

- My approach:

Intuition

Use the dictionary to save the count of every value in array

Code



```
length = len(nums)
dictionary = {}
for i in nums:
    if i in dictionary.keys():
        dictionary[i]+=1
    else:
        dictionary[i]=1
    if  dictionary[i]>length/2: return i
```

Complexity

Time: O(n)
Space: O(n)
Majority>n/2, so at most the #unique value = n - floor(n/2)

Key: the space complexity

217. Contains Duplicate

Sunday, June 16, 2019 10:09 PM

➢ Description

Link: https://leetcode.com/problems/contains-duplicate/

Given an array of integers, find if the array contains any duplicates.
Your function should return true if any value appears at least twice in the array, and it should return false if every element is distinct.

➢ Tag: ,

➢ Solution

- Approach1: Sort

Intuition

Complexity

Time: O(nlogn)
Space: O(1)

Key:

Code

```
def containsDuplicate(self, nums):
    """
    :type nums: List[int]
    :rtype: bool
    """
    nums.sort()
    for i in range(len(nums)-1):#注意边缘
        if nums[i]==nums[i+1]: return True
    return False
```

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(n)

Key: <Set> Data Structure (add, remove…)

Code

```
dic =set()
for i in nums:
    if i in dic:
        return True
    else:
        dic.add(i)
return False
```

- My approach

%219. Contains Duplicate II???

Sunday, June 16, 2019 11:33 PM

Description

Link: https://leetcode.com/problems/contains-duplicate-ii/

Given an array of integers and an integer k, find out whether there are two distinct indices i and j in the array such that nums[i] = nums[j] and the absolute difference between i and j is at most k.

Tag: Array

Solution

- Approach1:

Intuition

Complexity

Time: O(n^2)
Space: O(n)

Code

Key: enumerate: transfer to iterator

```python
class Solution(object):
    def containsNearbyDuplicate(self, nums, k):
        """
        :type nums: List[int]
        :type k: int
        :rtype: bool
        """
        dic = {}
        for i, v in enumerate(nums):
            if v in dic and i - dic[v] <= k:
                return True
            dic[v] = i
        return False
```

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

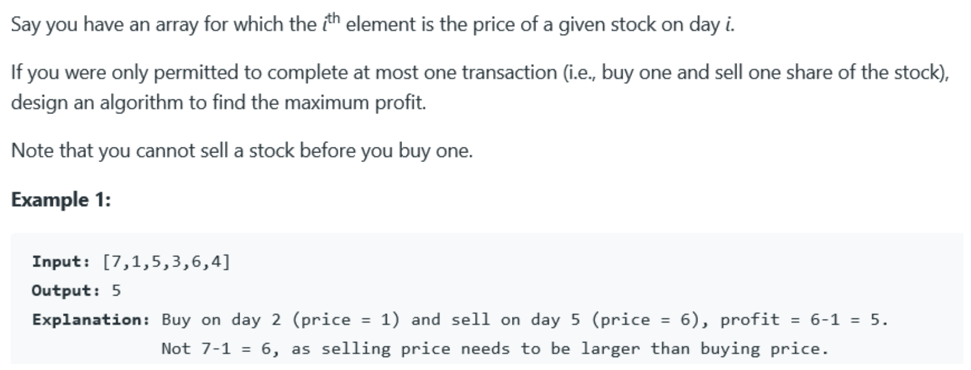
- My approach

*121.Best Time to Buy and Sell Stock

Tuesday, June 18, 2019 1:19 PM

➢ Description

Link: https://leetcode.com/problems/best-time-to-buy-and-sell-stock/



Say you have an array for which the i^{th} element is the price of a given stock on day *i*.

If you were only permitted to complete at most one transaction (i.e., buy one and sell one share of the stock), design an algorithm to find the maximum profit.

Note that you cannot sell a stock before you buy one.

Example 1:

```
Input: [7,1,5,3,6,4]
Output: 5
Explanation: Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = 6-1 = 5.
             Not 7-1 = 6, as selling price needs to be larger than buying price.
```

➢ Tag: ,Array, Maximum Subarray

➢ Solution

• Approach1: dynamic programming

Intuition

Save the valley of the price, and the max profit

Complexity

Time: O(n)
Space: O(1)

Key:

Code

```
minprice =100000000
maxprice =0
for i in range(len(prices)):
    if prices[i]<minprice:
        minprice = prices[i]
    elif prices[i]-minprice>maxprice:
        maxprice = prices[i]-minprice
return maxprice
```

• Approach2: My approach

Intuition

转化成最大子序列和问题

Complexity

Time: O(n)
Space: O(n)

Key:

Code

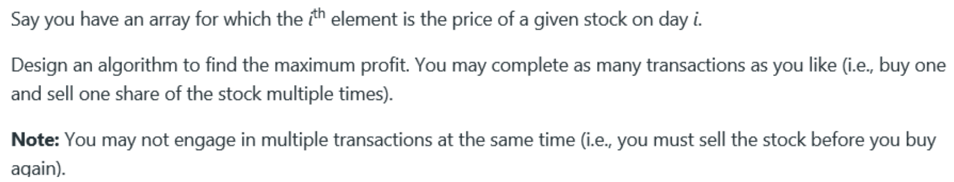
```
new = []
for i in range(len(prices)-1):
    new.append(prices[i+1]-prices[i])
thisMax= 0#max in this sequence
nowMax = 0#max in history
for j in range(len(new)):
    thisMax = max(new[j], new[j]+thisMax,0)
    nowMax = max(thisMax,nowMax)
return nowMax
```

*122. Best Time to Buy and Sell Stock II

Tuesday, June 18, 2019 2:47 PM

Description

Link: https://leetcode.com/problems/best-time-to-buy-and-sell-stock-ii/


Say you have an array for which the *i*th element is the price of a given stock on day *i*.

Design an algorithm to find the maximum profit. You may complete as many transactions as you like (i.e., buy one and sell one share of the stock multiple times).

Note: You may not engage in multiple transactions at the same time (i.e., you must sell the stock before you buy again).

Tag: ,Array

Solution

- Approach1:

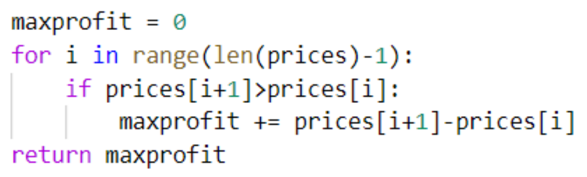
Intuition

As long as next one is larger than this one, we can accumulate profit

Complexity

Time: O(n)
Space: O(1)

Code


```
maxprofit = 0
for i in range(len(prices)-1):
    if prices[i+1]>prices[i]:
        maxprofit += prices[i+1]-prices[i]
return maxprofit
```

Key: 考虑如果prices = []

- Approach2:

Intuition

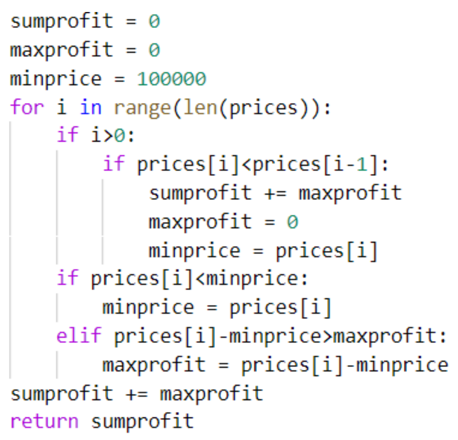
Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach


```
sumprofit = 0
maxprofit = 0
minprice = 100000
for i in range(len(prices)):
    if i>0:
        if prices[i]<prices[i-1]:
            sumprofit += maxprofit
            maxprofit = 0
            minprice = prices[i]
    if prices[i]<minprice:
        minprice = prices[i]
    elif prices[i]-minprice>maxprofit:
        maxprofit = prices[i]-minprice
sumprofit += maxprofit
return sumprofit
```

*53. Maximum Subarray

Tuesday, June 18, 2019 3:22 PM

- **Description**

  Link: https://leetcode.com/problems/maximum-subarray/

- **Tag: ,Array**
- **Solution**

- Approach1:

Intuition

Update nums, nums[i] means: so far maximum sum of continuous subarray.

Complexity

Time: O(n)
Space: O(1)

Key:

Code

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

- Approach2: Divide and Conquer

Intuition

(1)Max array on the left(2)max array on the right(3)max array between left and right: max array end with the middle + max array start with the middle

Complexity

Time: O(nlogn)
Space: O(1)

Key:

Code

```
def maxSubArrayHelper(self,nums, l, r):
    if l > r:
        return -2147483647
    m = (l+r) / 2

    leftMax = sumNum = 0
    for i in range(m - 1, l - 1, -1):
        sumNum += nums[i]
        leftMax = max(leftMax, sumNum)

    rightMax = sumNum = 0
    for i in range(m + 1, r + 1):
        sumNum += nums[i]
        rightMax = max(rightMax, sumNum)

    leftAns = self.maxSubArrayHelper(nums, l, m - 1)
    rightAns = self.maxSubArrayHelper(nums, m + 1, r)

    return max(leftMax + nums[m] + rightMax, max(leftAns, rightAns))

def maxSubArray(self, nums):
    return self.maxSubArrayHelper(nums, 0, len(nums) - 1)
```

- My approach: Dynamic

```
historymax = nums[0]
thismax = 0
for i in range(len(nums)):
    thismax = max(nums[i],nums[i]+thismax)
    historymax = max(thismax, historymax)
return historymax
```

Key: when you need to set a variable to save max/min value, think about it initialize as 0 or the first value in the array or the min/max number in system.

88. Merge Sorted Array

Tuesday, June 18, 2019 3:54 PM

- **Description**

  Link: https://leetcode.com/problems/merge-sorted-array/

  Given two sorted integer arrays *nums1* and *nums2*, merge *nums2* into *nums1* as one sorted array.

  Note:

  - The number of elements initialized in *nums1* and *nums2* are *m* and *n* respectively.
  - You may assume that *nums1* has enough space (size that is greater or equal to *m* + *n*) to hold additional elements from *nums2*.

- **Tag: ,Array**
- **Solution**

- Approach1:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

```
def merge(self, nums1, m, nums2, n):
        while m > 0 and n > 0:
            if nums1[m-1] >= nums2[n-1]:
                nums1[m+n-1] = nums1[m-1]
                m -= 1
            else:
                nums1[m+n-1] = nums2[n-1]
                n -= 1
        if n > 0: # if nums2 left
            nums1[:n] = nums2[:n]
```

Key:

- My approach

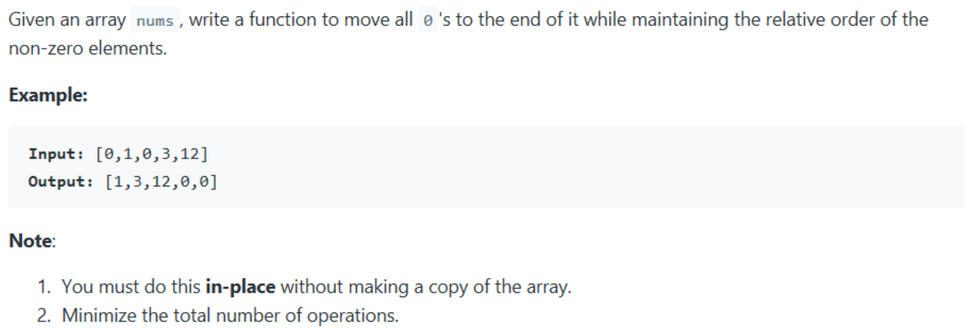
```
i = m-1
j = n-1
t = -1
while(j>-1):
    if i<0: # if nums2 left
        nums1[t] = nums2[j]
        j -= 1
        t -= 1
    elif nums1[i]>nums2[j]:
        nums1[t] = nums1[i]
        i -= 1
        t -=1
    elif nums1[i]<=nums2[j]:
        nums1[t] = nums2[j]
        j -= 1
        t -= 1

return nums1
```

283. Move Zeroes

Tuesday, June 18, 2019   4:25 PM

Description

Link: https://leetcode.com/problems/move-zeroes/



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

Example:

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

Note:

1. You must do this **in-place** without making a copy of the array.
2. Minimize the total number of operations.

Tag: Array

Solution

- Approach1:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- Approach2:

Intuition

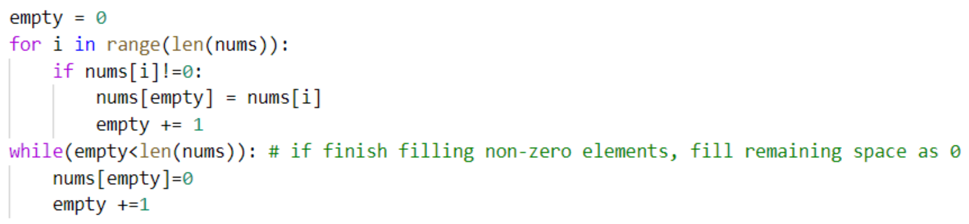
Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach



```
empty = 0
for i in range(len(nums)):
    if nums[i]!=0:
        nums[empty] = nums[i]
        empty += 1
while(empty<len(nums)): # if finish filling non-zero elements, fill remaining space as 0
    nums[empty]=0
    empty +=1
```

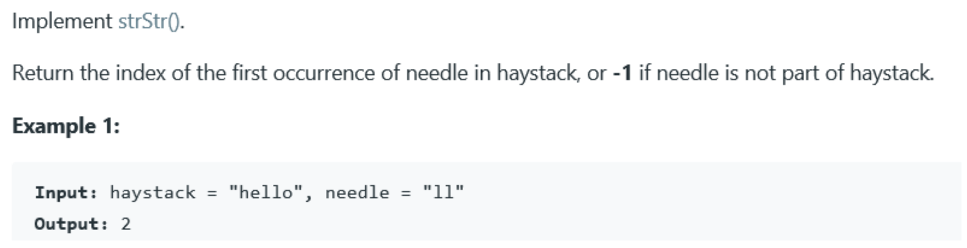
Time: O(n)
Space: O(1)

#28. Implement strStr()

Tuesday, June 18, 2019 5:23 PM

➢ Description

Link: https://leetcode.com/problems/implement-strstr/



Implement strStr().

Return the index of the first occurrence of needle in haystack, or **-1** if needle is not part of haystack.

Example 1:

```
Input: haystack = "hello", needle = "ll"
Output: 2
```

➢ Tag: ,String

➢ Solution

- Approach1:

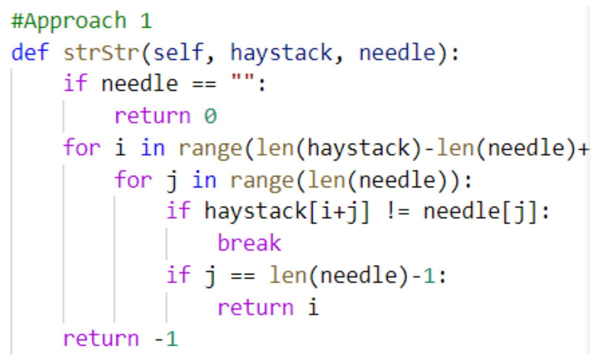
Intuition

Complexity

Time: O(n*m)
Space: O(1)

Key: Interview: What if needle="", return 0

Code



```
#Approach 1
def strStr(self, haystack, needle):
    if needle == "":
        return 0
    for i in range(len(haystack)-len(needle)+
        for j in range(len(needle)):
            if haystack[i+j] != needle[j]:
                break
            if j == len(needle)-1:
                return i
    return -1
```

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

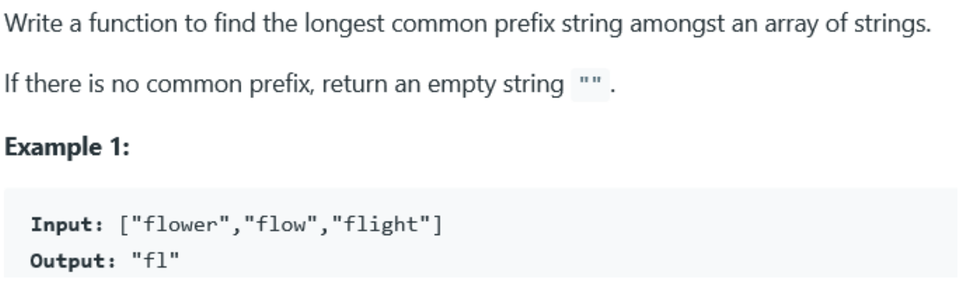
- My approach

#14. Longest Common Prefix

Tuesday, June 18, 2019 8:55 PM

➢ Description

Link: https://leetcode.com/problems/longest-common-prefix/



Write a function to find the longest common prefix string amongst an array of strings.

If there is no common prefix, return an empty string `""`.

Example 1:

```
Input: ["flower","flow","flight"]
Output: "fl"
```

➢ Tag: , String

➢ Solution

- Approach1: vertical

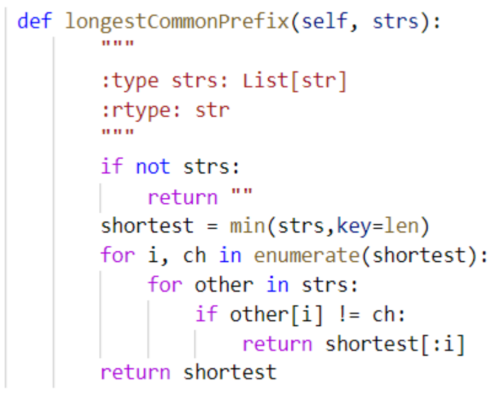
Intuition

注意 strs = [] ,strs=[""]

Complexity

Time: O(n)
Space: O(1)

Code



```
def longestCommonPrefix(self, strs):
        """
        :type strs: List[str]
        :rtype: str
        """
        if not strs:
            return ""
        shortest = min(strs,key=len)
        for i, ch in enumerate(shortest):
            for other in strs:
                if other[i] != ch:
                    return shortest[:i]
        return shortest
```

Key: string切片str[:4]

- Approach2:

Intuition

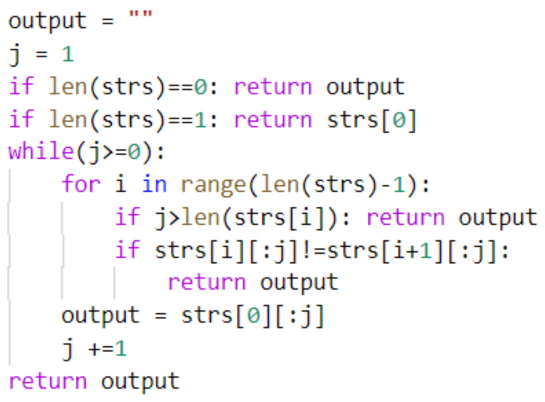
Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach



```python
output = ""
j = 1
if len(strs)==0: return output
if len(strs)==1: return strs[0]
while(j>=0):
    for i in range(len(strs)-1):
        if j>len(strs[i]): return output
        if strs[i][:j]!=strs[i+1][:j]:
            return output
    output = strs[0][:j]
    j +=1
return output
```

#58. Length of Last Word

Tuesday, June 18, 2019 9:34 PM

Description

Link: https://leetcode.com/problems/length-of-last-word/

Given a string *s* consists of upper/lower-case alphabets and empty space characters `' '`, return the length of last word in the string.

If the last word does not exist, return 0.

Note: A word is defined as a character sequence consists of non-space characters only.

Tag: , String

Solution

- Approach1:

Intuition

"a", "a ", " a " "", " "," "

Complexity

Time: O(n)
Space: O(1)

Code

Key:

```
cnt = 0#save count of last nonspace character
for v in reversed(s):
    if v.isspace():
        if cnt: break #if !=0, return count
    else: cnt += 1
return cnt
```

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach

```
if len(s)==0: return 0
nonspace = 0
space = 0
for i in range(len(s)-1,-1,-1):
    if s[i]==" ":
        if nonspace==0:
            space+=1
            continue
        else:
            return len(s)-i-1-space
    else:
        nonspace+=1
if nonspace!=0: return len(s)-space
else: return 0
```

387. First Unique Character in a String

Tuesday, June 18, 2019 10:57 PM

- **Description**

Link: https://leetcode.com/problems/first-unique-character-in-a-string/

Given a string, find the first non-repeating character in it and return it's index. If it doesn't exist, return -1.

Examples:

```
s = "leetcode"
return 0.
```

- **Tag: , String**
- **Solution**

- Approach1:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key: str.index("a")

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach

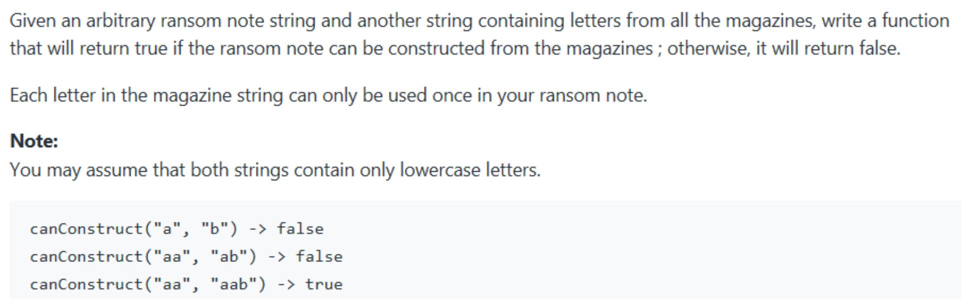
```
def firstUniqChar(self, s: str) -> int:
    dic ={}
    idx ={}
    for i in range(len(s)):
        if s[i] in dic.keys():
            dic[s[i]]+=1
        else:
            dic[s[i]] = 1
            idx[s[i]] = i
    for j in dic.keys():
        if dic[j]==1:
            return idx[j]
    return -1
```

383. Ransom Note

Tuesday, June 18, 2019   11:05 PM

Description

Link: https://leetcode.com/problems/ransom-note/



Given an arbitrary ransom note string and another string containing letters from all the magazines, write a function that will return true if the ransom note can be constructed from the magazines ; otherwise, it will return false.

Each letter in the magazine string can only be used once in your ransom note.

Note:
You may assume that both strings contain only lowercase letters.

```
canConstruct("a", "b") -> false
canConstruct("aa", "ab") -> false
canConstruct("aa", "aab") -> true
```

Tag: , String

Solution

- Approach1:

Intuition

Complexity

Time: O(n+m)
Space: O(26)

Code

Key: set(),count()

```
for i in set(ransomNote):
    if ransomNote.count(i) > magazine.count(i):
        return False
return True
```

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach

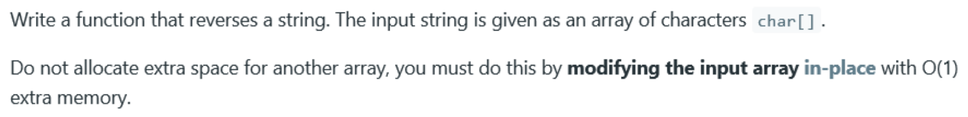
```
dic = {}
for m in magazine:
    if m in dic.keys():
        dic[m]+=1
    else:
        dic[m]=1
for r in ransomNote:
    if r in dic.keys():
        dic[r] -=1
        if dic[r]<0:
            return False
    else:
        return False
return True
```

344. Reverse String

Wednesday, June 19, 2019 12:59 AM

- **Description**

Link: https://leetcode.com/problems/reverse-string/



Write a function that reverses a string. The input string is given as an array of characters `char[]`.

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

- **Tag: ,Array**
- **Solution**

- Approach1:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- Approach2:

Intuition

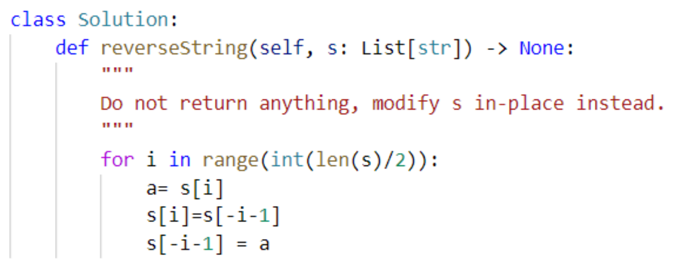
Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach



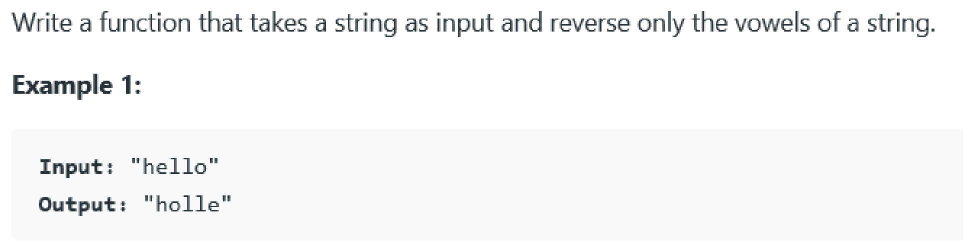
```
class Solution:
    def reverseString(self, s: List[str]) -> None:
        """
        Do not return anything, modify s in-place instead.
        """
        for i in range(int(len(s)/2)):
            a= s[i]
            s[i]=s[-i-1]
            s[-i-1] = a
```

345. Reverse Vowels of a String

Wednesday, June 19, 2019 11:18 AM

Description

Link: https://leetcode.com/problems/reverse-vowels-of-a-string/


Write a function that takes a string as input and reverse only the vowels of a string.

Example 1:

```
Input: "hello"
Output: "holle"
```

Tag: , String, Two pointers

Solution

- Approach1: My approach

Intuition

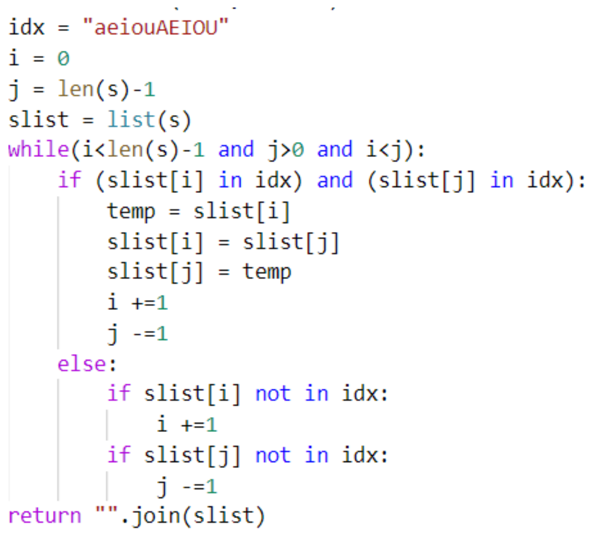
Both upper case and lower case

Complexity

Time: O(n)
Space: O(1)

Key: 'str' object does not support item assignment in Python: if you want to change the character in the str, list(str)-->change-->"".join(list)

Code


```
idx = "aeiouAEIOU"
i = 0
j = len(s)-1
slist = list(s)
while(i<len(s)-1 and j>0 and i<j):
    if (slist[i] in idx) and (slist[j] in idx):
        temp = slist[i]
        slist[i] = slist[j]
        slist[j] = temp
        i +=1
        j -=1
    else:
        if slist[i] not in idx:
            i +=1
        if slist[j] not in idx:
            j -=1
return "".join(slist)
```

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

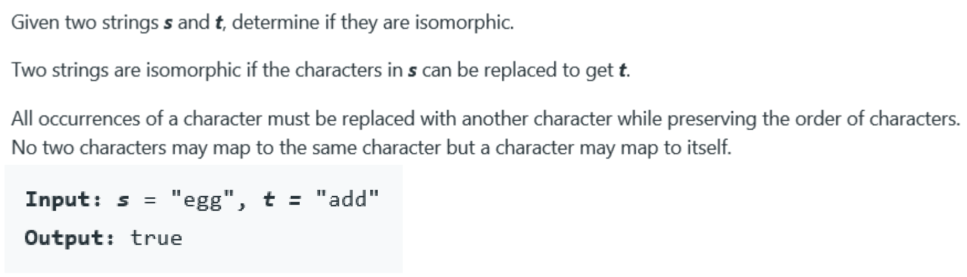
- My approach

#205. Isomorphic Strings

Wednesday, June 19, 2019 11:26 AM

Description

Link: https://leetcode.com/problems/isomorphic-strings/



Given two strings ***s*** and ***t***, determine if they are isomorphic.

Two strings are isomorphic if the characters in ***s*** can be replaced to get ***t***.

All occurrences of a character must be replaced with another character while preserving the order of characters. No two characters may map to the same character but a character may map to itself.

```
Input: s = "egg", t = "add"
Output: true
```

Tag: , String

Solution

- Approach1:My approach Dictionary

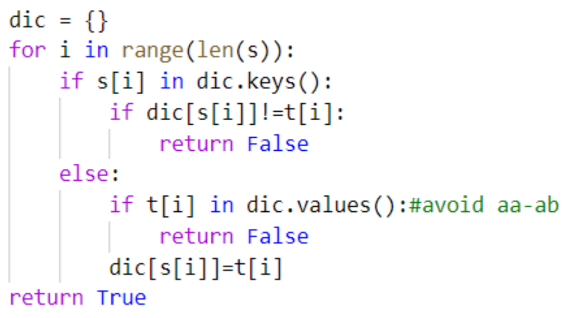
Intuition

Paper-title, ab-aa

Complexity

Time: O(n)
Space: O(1)

Code



```
dic = {}
for i in range(len(s)):
    if s[i] in dic.keys():
        if dic[s[i]]!=t[i]:
            return False
    else:
        if t[i] in dic.values():#avoid aa-ab
            return False
        dic[s[i]]=t[i]
return True
```

Key:

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

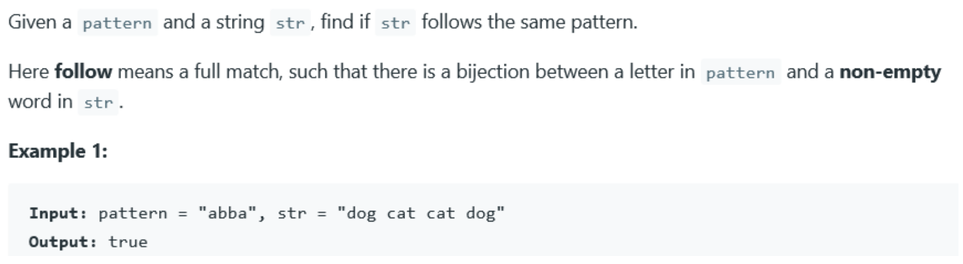
- My approach

#290. Word Pattern

Wednesday, June 19, 2019 12:28 PM

Description

Link: https://leetcode.com/problems/word-pattern/



Given a `pattern` and a string `str`, find if `str` follows the same pattern.

Here **follow** means a full match, such that there is a bijection between a letter in `pattern` and a **non-empty** word in `str`.

Example 1:

```
Input: pattern = "abba", str = "dog cat cat dog"
Output: true
```

Tag: , String

Solution

- Approach1:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key: See 205;str.split()

```
dic={}
str = str.split()
if len(pattern)!=len(str):
    return False
for i in range(len(pattern)):
    word = str[i]
    if pattern[i] in dic.keys():
        if dic[pattern[i]]!=word:
            return False
    else:
        if word in dic.values():#avoid abba--dog dog dog dog
            return False
        dic[pattern[i]] = word
return True
```

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach

242. Valid Anagram

Wednesday, June 19, 2019 12:41 PM

➢ Description

Link: https://leetcode.com/problems/valid-anagram/

Given two strings *s* and *t* , write a function to determine if *t* is an anagram of *s*.

Example 1:

```
Input: s = "anagram", t = "nagaram"
Output: true
```

➢ Tag: ,

➢ Solution

- Approach1:My approach Dictionary

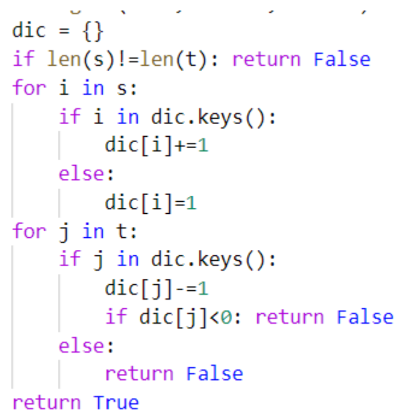
Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:



```
dic = {}
if len(s)!=len(t): return False
for i in s:
    if i in dic.keys():
        dic[i]+=1
    else:
        dic[i]=1
for j in t:
    if j in dic.keys():
        dic[j]-=1
        if dic[j]<0: return False
    else:
        return False
return True
```

- Approach2: sort

Intuition

Complexity

Time: O(nlogn)
Space: O(n)

Code

Key:

- My approach

38. Count and Say

Wednesday, June 19, 2019 1:16 PM

➢ Description

Link: https://leetcode.com/problems/count-and-say/

The count-and-say sequence is the sequence of integers with the first five terms as following:

```
1.     1
2.     11
3.     21
4.     1211
5.     111221
```

➢ Tag: , String

➢ Solution

- Approach1:My approach

Intuition

注意n和layer的对应

Complexity

Time: O(n)
Space: O(1)

Code

Key:

```
last ="1"#string on last layer
for i in range(n-1):   #Attention
    pre =""      #accumulate string
    prechar =last[0]     # repeat character
    count = 0      #number of repeat character
    for thischar in last:
        if thischar != prechar:
            pre = pre+str(count)+prechar     #concatenate
            count=0
            prechar = thischar
        if thischar==prechar:
            count +=1
    pre = pre+str(count)+prechar
    last = pre
return last
```

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach

%168. Excel Sheet Column Title

Wednesday, June 19, 2019 1:18 PM

Description

Link: https://leetcode.com/problems/excel-sheet-column-title/

```
1 -> A
2 -> B
3 -> C
...
26 -> Z
27 -> AA
28 -> AB
...
```

Tag: , String, 进制

Solution

- Approach1:

Intuition

层层取余

Complexity

Time: O(n)
Space: O(1)

Code

Key: ord(), chr(), //整数除

```
def convertToTitle(self, num):
    capitals = [chr(x) for x in range(ord('A'), ord('Z')+1)]
    result = []
    while num > 0:
        result.append(capitals[(num-1)%26])
        num = (num-1) // 26
    result.reverse()
    return ''.join(result)
```

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

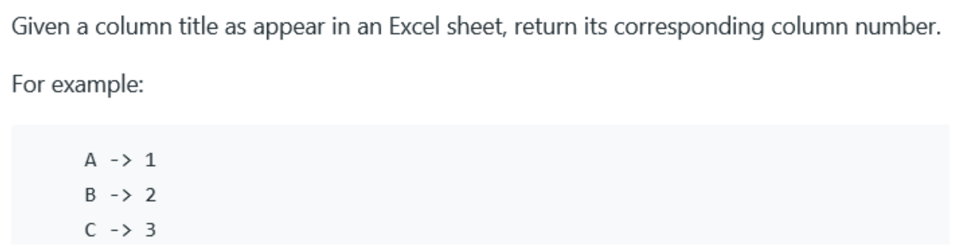
- My approach

171. Excel Sheet Column Number

Wednesday, June 19, 2019 2:15 PM

➢ Description

Link: https://leetcode.com/problems/excel-sheet-column-number/



Given a column title as appear in an Excel sheet, return its corresponding column number.

For example:

```
    A -> 1
    B -> 2
    C -> 3
```

➢ Tag: , string, 进制

➢ Solution

- Approach1:

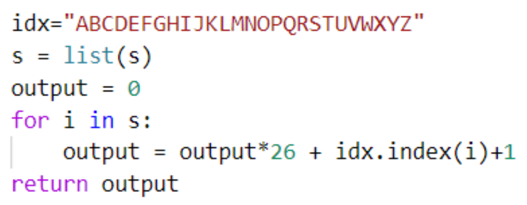
Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:



```
idx="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
s = list(s)
output = 0
for i in s:
    output = output*26 + idx.index(i)+1
return output
```

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach

*13. Roman to Integer

Wednesday, June 19, 2019 2:16 PM

➢ Description

Link: https://leetcode.com/problems/roman-to-integer/

➢ Tag: ,

➢ Solution

- Approach1:

Intuition

# *Note: The trick is that the last letter is always added.
# Except the last one, if one letter is less than its latter one, this letter is subtracted.

Complexity

Time: O(n)
Space: O(1)

Code

Key:

```
roman = {'M': 1000,'D': 500 ,'C': 100,'L': 50,'X': 10,'V': 5,'I': 1}
z = 0
for i in range(0, len(s) - 1):
    if roman[s[i]] < roman[s[i+1]]:
        z -= roman[s[i]]
    else:
        z += roman[s[i]]
return z + roman[s[-1]]
```

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach

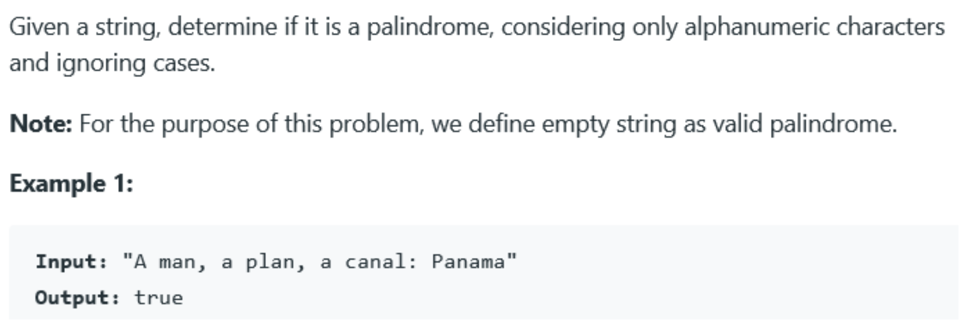
```
dic={'I':1,'V':5,'X':10,'L':50,'C':100,'D':500,'M':1000,'IV':4,'IX':9,'XL':40,'XC':90,'CD':400,'CM':900}
output = 0
i=0
while(i<len(s)):
    if s[i:i+2] in dic.keys():
        output += dic[s[i:i+2]]
        i +=2
        continue
    else:
        output += dic[s[i]]
        i +=1
return output
```

125. Valid Palindrome

Wednesday, June 19, 2019 2:38 PM

Description

Link: https://leetcode.com/problems/valid-palindrome/



Given a string, determine if it is a palindrome, considering only alphanumeric characters and ignoring cases.

Note: For the purpose of this problem, we define empty string as valid palindrome.

Example 1:

```
Input: "A man, a plan, a canal: Panama"
Output: true
```

Tag: , string, Two pointers, 回文

Solution

- Approach1:My approach

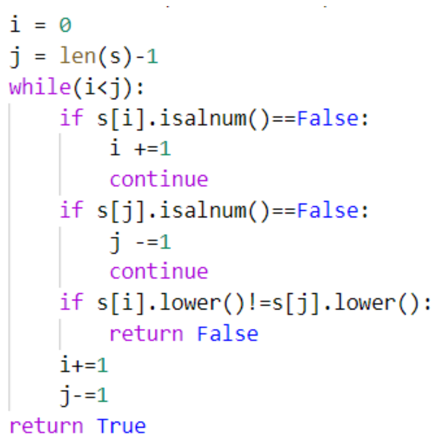
Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key: isalnum()



```
i = 0
j = len(s)-1
while(i<j):
    if s[i].isalnum()==False:
        i +=1
        continue
    if s[j].isalnum()==False:
        j -=1
        continue
    if s[i].lower()!=s[j].lower():
        return False
    i+=1
    j-=1
return True
```

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

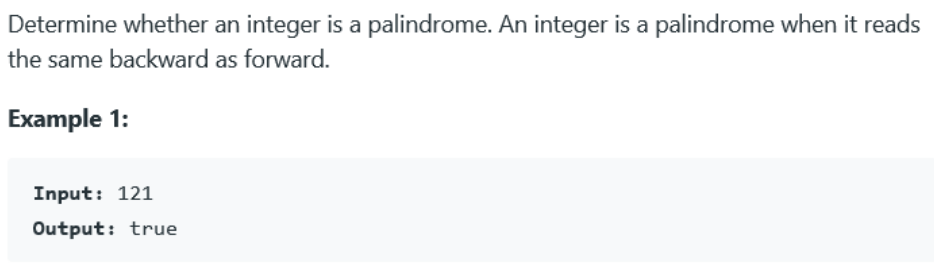
- My approach

9. Palindrome Number??

Wednesday, June 19, 2019 4:03 PM

Description

Link: https://leetcode.com/problems/palindrome-number/



Determine whether an integer is a palindrome. An integer is a palindrome when it reads the same backward as forward.

Example 1:

```
Input: 121
Output: true
```

Tag: ,

Solution

- Approach1:not convert to string

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- Approach2:

Intuition

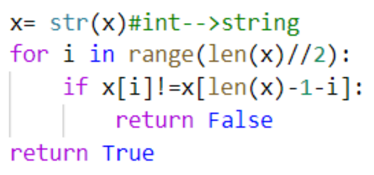
Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach



```
x= str(x)#int-->string
for i in range(len(x)//2):
    if x[i]!=x[len(x)-1-i]:
        return False
return True
```

#20. Valid Parentheses

Wednesday, June 19, 2019 4:14 PM

Description

Link: https://leetcode.com/problems/valid-parentheses/

Given a string containing just the characters `'('`, `')'`, `'{'`, `'}'`, `'['` and `']'`, determine if the input string is valid.

An input string is valid if:

1. Open brackets must be closed by the same type of brackets.
2. Open brackets must be closed in the correct order.

Note that an empty string is also considered valid.

Example 1:

```
Input: "()"
Output: true
```

Tag: ,String, Stack

Solution

- Approach1:My approach Stack

Intuition

Complexity

Time: O(n)
Space: O(n)
Worse case save everything in stack

Key: 注意考虑各种条件

Code

```
stack = []
dic={')':'(',']':'[','}':'{'}
for i in s:
    if i=='(' or i=='[' or i=='{':
        stack.append(i)
    else:
        if len(stack)==0 or stack[-1]!=dic[i]:  #'}' or '(]'
            return False
        else:
            stack.pop(-1)
if len(stack)!=0:return False   #'('
return True #'' or '()'
#  return stack == []
```

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach

%100. Same Tree

Wednesday, June 19, 2019 4:30 PM

➢ Description

Link: https://leetcode.com/problems/same-tree/

Given two binary trees, write a function to check if they are the same or not.

Two binary trees are considered the same if they are structurally identical and the nodes have the same value.

➢ Tag: ,Tree, DFS&BFS, Stack& Queue

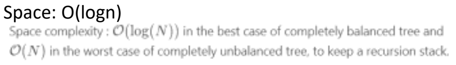
➢ Solution

- Approach1: Recursive

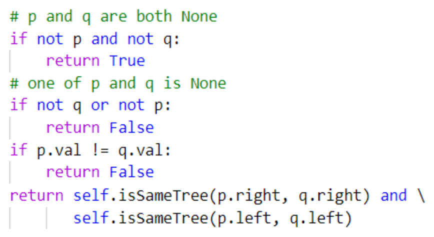
Intuition

Complexity

Time: O(n)
Space: O(logn)


Space complexity : $\mathcal{O}(\log(N))$ in the best case of completely balanced tree and $\mathcal{O}(N)$ in the worst case of completely unbalanced tree, to keep a recursion stack.

Code



```python
# p and q are both None
if not p and not q:
    return True
# one of p and q is None
if not q or not p:
    return False
if p.val != q.val:
    return False
return self.isSameTree(p.right, q.right) and \
       self.isSameTree(p.left, q.left)
```

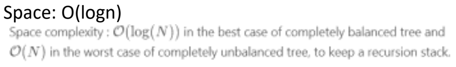
Key:

- Approach2: DFS with stack

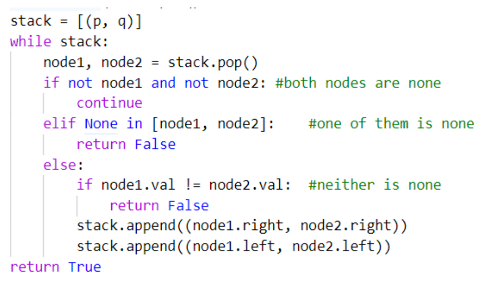
Intuition

Complexity

Time: O(n)
Space: O(logn)


Space complexity : $\mathcal{O}(\log(N))$ in the best case of completely balanced tree and $\mathcal{O}(N)$ in the worst case of completely unbalanced tree, to keep a recursion stack.

Code



```python
stack = [(p, q)]
while stack:
    node1, node2 = stack.pop()
    if not node1 and not node2: #both nodes are none
        continue
    elif None in [node1, node2]:    #one of them is none
        return False
    else:
        if node1.val != node2.val:  #neither is none
            return False
        stack.append((node1.right, node2.right))
        stack.append((node1.left, node2.left))
return True
```

Key:

- Approach3: BFS with queue

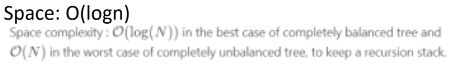
Wednesday, June 19, 2019 5:49 PM

- ⭐ Approach3: BFS with queue

Intuition

Complexity

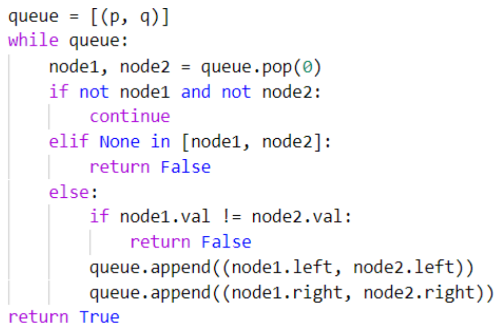
Time: O(n)
Space: O(logn)



Space complexity : $\mathcal{O}(\log(N))$ in the best case of completely balanced tree and $\mathcal{O}(N)$ in the worst case of completely unbalanced tree, to keep a recursion stack.

? **Key**: What is balance tree

Code



```
queue = [(p, q)]
while queue:
    node1, node2 = queue.pop(0)
    if not node1 and not node2:
        continue
    elif None in [node1, node2]:
        return False
    else:
        if node1.val != node2.val:
            return False
        queue.append((node1.left, node2.left))
        queue.append((node1.right, node2.right))
return True
```

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach

%101. Symmetric Tree??

Wednesday, June 19, 2019 7:02 PM

- **Description**

  Link:

- **Tag: ,**
- **Solution**

- Approach1:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

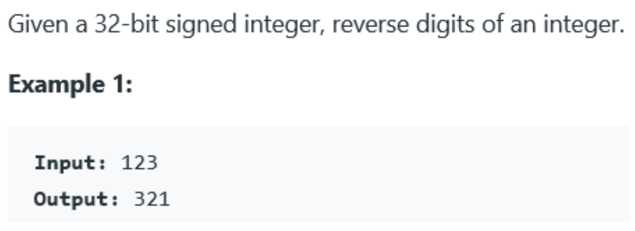
- My approach

%7. Reverse Integer

Wednesday, June 19, 2019 6:36 PM

➢ Description

Link: https://leetcode.com/problems/reverse-integer/



Given a 32-bit signed integer, reverse digits of an integer.

Example 1:

```
Input: 123
Output: 321
```

➢ Tag: , 进制

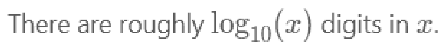
➢ Solution

- Approach1:

Intuition

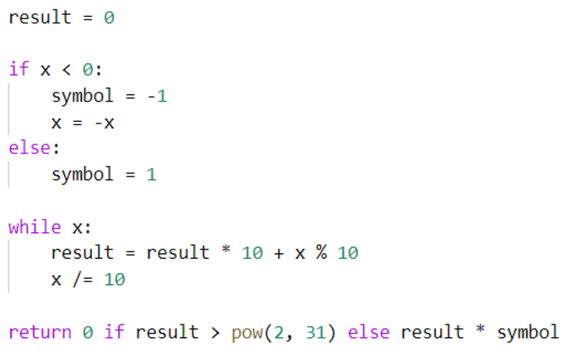
Complexity

Time: O(logn)
Space: O(1)



There are roughly $\log_{10}(x)$ digits in x.

Code



```
result = 0

if x < 0:
    symbol = -1
    x = -x
else:
    symbol = 1

while x:
    result = result * 10 + x % 10
    x /= 10

return 0 if result > pow(2, 31) else result * symbol
```

Key:

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach

#66.Plus One

Wednesday, June 19, 2019 6:59 PM

Description

Link: https://leetcode.com/problems/plus-one/

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

Tag: , 进制

Solution

- Approach1: digits-->integer

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

```
num = 0
for i in range(len(digits)):
    num += digits[i] * pow(10, (len(digits)-1-i))
return [int(i) for i in str(num+1)]
```

Key:

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach: Recursion

```
if len(digits)==0: return [1]
if digits[-1]==9:
    digits = self.plusOne(digits[0:-1])
    digits.append(0)
else:
    digits[-1]+=1
return digits
```

*258. Add Digits

Wednesday, June 19, 2019 7:05 PM

Description

Link: https://leetcode.com/problems/add-digits/

Given a non-negative integer `num`, repeatedly add all its digits until the result has only one digit.

Example:

```
Input: 38
Output: 2
Explanation: The process is like: 3 + 8 = 11, 1 + 1 = 2.
             Since 2 has only one digit, return it.
```

No loop/recursion

Tag: , 进制

Solution

- Approach1: Loop

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

```
while(num >= 10):
    temp = 0
    while(num > 0):
        temp += num % 10
        num /= 10
    num = temp
return num
```

- Approach2: Digital root

Intuition

this method depends on the truth:

N=(a[0] * 1 + a[1] * 10 + ...a[n] * 10 ^n),and a[0]...a[n] are all between [0,9]

we set M = a[0] + a[1] + ..a[n]

and another truth is that:

1 % 9 = 1

10 % 9 = 1

100 % 9 = 1

so N % 9 = a[0] + a[1] + ..a[n]

means N % 9 = M

so N = M (% 9)

as 9 % 9 = 0,so we can make (n - 1) % 9 + 1 to help us solve the problem when n is 9.as N is 9, (9 - 1) % 9 + 1 = 9

Complexity

Time: O(n)
Space: O(1)

Code

Key:

```
if num == 0 : return 0
else:return (num - 1) % 9 + 1
```

- My approach

*67. Add Binary

Wednesday, June 19, 2019 7:31 PM

- **Description**

Link: https://leetcode.com/problems/add-binary/submissions/

Given two binary strings, return their sum (also a binary string).

The input strings are both **non-empty** and contains only characters `1` or `0`.

Example 1:

```
Input: a = "11", b = "1"
Output: "100"
```

- **Tag: , 进制**
- **Solution**

- Approach1:

Intuition

Complexity

Time: O(max(a,b))
Space: O(n)

Code

```
res, carry = '', 0
i, j = len(a) - 1, len(b) - 1
while i >= 0 or j >= 0 or carry:
    curval = (i >= 0 and a[i] == '1') + (j >= 0 and b[j] == '1')
    carry, rem = divmod(curval + carry, 2)
    res = str(rem) + res
    i -= 1
    j -= 1
return res
```

Key: divmod() 函数把除数和余数运算结果结合起来，返回一个包含商和余数的元组(a // b, a % b)。

- Approach2:Recursion

Intuition

Complexity

Time: O(??)
Space: O(1)

Key:

Code

```
if len(a)==0: return b
if len(b)==0: return a
if a[-1] == '1' and b[-1] == '1':
    return self.addBinary(self.addBinary(a[0:-1],b[0:-1]),'1')+'0'
if a[-1] == '0' and b[-1] == '0':
    return self.addBinary(a[0:-1],b[0:-1])+'0'
else:
    return self.addBinary(a[0:-1],b[0:-1])+'1'
```

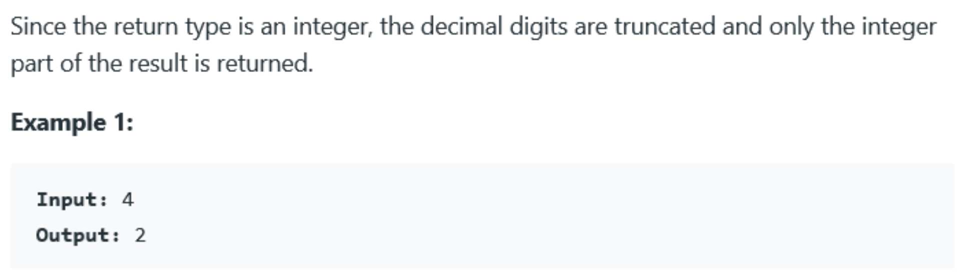
- My approach

%69. Sqrt(x)

Friday, June 21, 2019 2:26 PM

Description

Link: https://leetcode.com/problems/sqrtx/



Since the return type is an integer, the decimal digits are truncated and only the integer part of the result is returned.

Example 1:

```
Input: 4
Output: 2
```

Tag: , Binary Search

Solution

- Approach1:

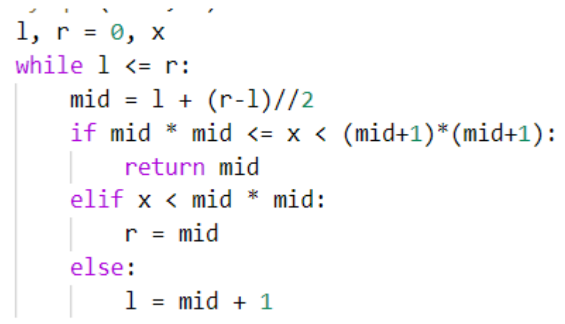
Intuition

Complexity

Time: O(logn)
Space: O(1)

Code

Key:



```
l, r = 0, x
while l <= r:
    mid = l + (r-l)//2
    if mid * mid <= x < (mid+1)*(mid+1):
        return mid
    elif x < mid * mid:
        r = mid
    else:
        l = mid + 1
```

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach

367. Valid Perfect Square

Friday, June 21, 2019 7:13 PM

Description

Link: https://leetcode.com/problems/valid-perfect-square/

Given a positive integer *num*, write a function which returns True if *num* is a perfect square else False.

Tag: , Binary Search

Solution

- Approach1: My approach

Intuition

Complexity

Time: O(logn)
Space: O(1)

Code

Key: Two pointers

```
if num==1: return True
low = 0
high = num
mid = num//2
while(mid>1):
    if mid*mid>num:
        high = mid
    elif mid*mid<num:
        low = mid
    else:
        return  True
    if (high-low == 1): return False
    mid = (low+high)//2
return False
```

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach

%204. Count Primes

Wednesday, July 3, 2019 6:43 PM

Description

Link: https://leetcode.com/problems/count-primes/

Count the number of prime numbers less than a non-negative number, ***n***.

Tag: ,

Solution

- Approach1:

Intuition

Complexity

Time: O(n^2)
Space: O(1)

Code

Key: isPrime() function;

- Approach2:

Intuition
https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes

Complexity

Time: O(nloglogn)
Space: O(1)

Key: Sieve of Eratosthenes

Code

```
if n <= 2:
    return 0
res = [True] * n
res[0] = res[1] = False
for i in range(2, n):
    if res[i] == True:
        for j in range(2, (n-1)//i+1):
            res[i*j] = False
return sum(res)
```

- My approach

%1. Two Sum

Wednesday, July 3, 2019 6:41 PM

Description

Link: https://leetcode.com/problems/two-sum/

```
Given nums = [2, 7, 11, 15], target = 9,

Because nums[0] + nums[1] = 2 + 7 = 9,
return [0, 1].
```

Tag: ,

Solution

- Approach1:

Intuition

Complexity

Time: O(n)
Space: O(1)

Key: dictionary

Code

```
dic ={}
for i,n  in enumerate(nums):
    if target-n in dic:
        return [dic[target-n],i]
    else:
        dic[n]=i
```

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach

*167. Two Sum II - Input array is sorted

Wednesday, July 3, 2019 7:41 PM

➢ Description

Link: https://leetcode.com/problems/two-sum-ii-input-array-is-sorted/

➢ Tag: , Two pointers

➢ Solution

- Approach1: Two pointers

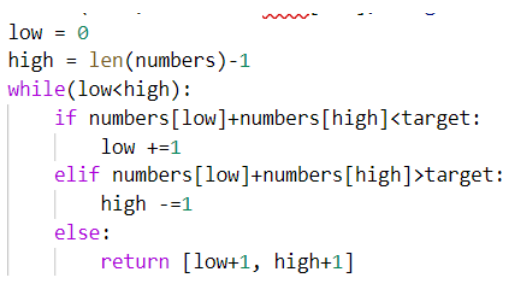
Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key: 可以为负数



```
low = 0
high = len(numbers)-1
while(low<high):
    if numbers[low]+numbers[high]<target:
        low +=1
    elif numbers[low]+numbers[high]>target:
        high -=1
    else:
        return [low+1, high+1]
```

- Approach2: Dictionary

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

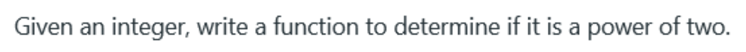
- My approach

*231. Power of Two

Wednesday, July 3, 2019 8:38 PM

Description

Link: https://leetcode.com/problems/power-of-two/



Given an integer, write a function to determine if it is a power of two.

Tag: , Bit Operation

Solution

- ### Approach1: Bit Count

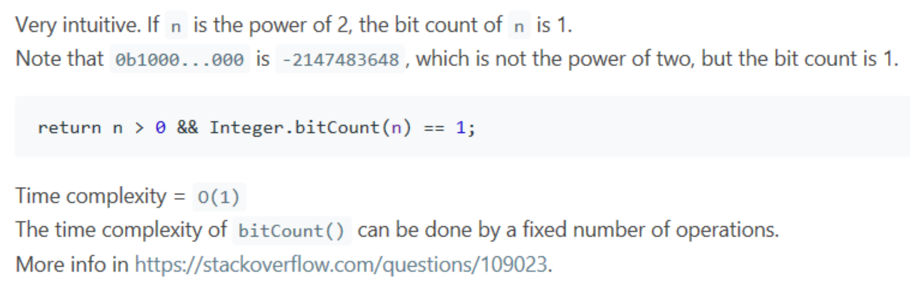
Intuition

Complexity

Time: O(1)
Space: O(1)

Code

Key:



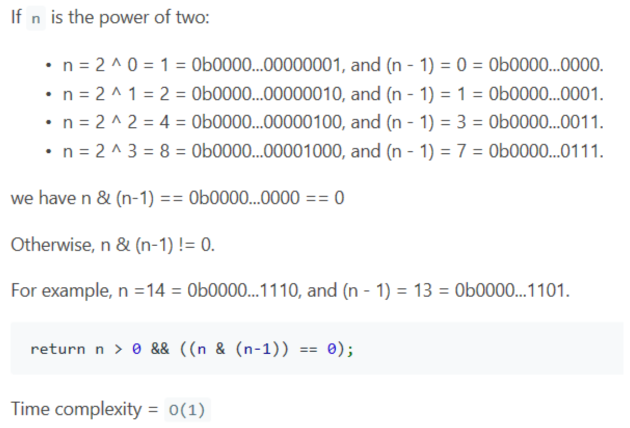
Very intuitive. If `n` is the power of 2, the bit count of `n` is 1.
Note that `0b1000...000` is `-2147483648`, which is not the power of two, but the bit count is 1.

```
return n > 0 && Integer.bitCount(n) == 1;
```

Time complexity = `O(1)`
The time complexity of `bitCount()` can be done by a fixed number of operations.
More info in https://stackoverflow.com/questions/109023.

- ### Approach2: Bit Operation

Intuition



If `n` is the power of two:

- n = 2 ^ 0 = 1 = 0b0000...00000001, and (n - 1) = 0 = 0b0000...0000.
- n = 2 ^ 1 = 2 = 0b0000...00000010, and (n - 1) = 1 = 0b0000...0001.
- n = 2 ^ 2 = 4 = 0b0000...00000100, and (n - 1) = 3 = 0b0000...0011.
- n = 2 ^ 3 = 8 = 0b0000...00001000, and (n - 1) = 7 = 0b0000...0111.

we have n & (n-1) == 0b0000...0000 == 0

Otherwise, n & (n-1) != 0.

For example, n =14 = 0b0000...1110, and (n - 1) = 13 = 0b0000...1101.

```
return n > 0 && ((n & (n-1)) == 0);
```

Time complexity = `O(1)`

Complexity

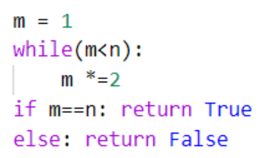
Time: O(1)
Space: O(1)

Num & (-num)==num

Code

Key:

- ### My approach



```
m = 1
while(m<n):
    m *=2
if m==n: return True
else: return False
```

O(logn)

*326. Power of Three

Wednesday, July 3, 2019 8:47 PM

Description

Link: https://leetcode.com/problems/power-of-three/

Tag: , Math

Solution

- Approach1: Math

Intuition

We can use mathematics as follows

$$n = 3^i$$
$$i = \log_3(n)$$
$$i = \frac{\log_b(n)}{\log_b(3)}$$

`n` is a power of three if and only if `i` is an integer. In Java, we check if a number is an integer by taking the decimal part (using `% 1`) and checking if it is 0.

Complexity

Time: O(1)
Space: O(1)

Code

```
return n > 0 and abs(math.log(n, 3) - round(math.log(n, 3))) < 1e-10
```

Key:

- Approach2: Integer limitations

Intuition

$$3^{\lfloor \log_3 MaxInt \rfloor} = 3^{\lfloor 19.56 \rfloor} = 3^{19} = 1162261467$$

Therefore, the possible values of `n` where we should return `true` are 3^0, 3^1 ... 3^{19}. Since 3 is a prime number, the only divisors of 3^{19} are 3^0, 3^1 ... 3^{19}, therefore all we need to do is divide 3^{19} by `n`. A remainder of **0** means `n` is a divisor of 3^{19} and therefore a power of three.

Complexity

Time: O(1)
Space: O(1)

Code

```
return n > 0 and 1162261467 % n == 0
```

Key:

- My approach

Same as power of two log3(n)

%342. Power of Four???

Friday, July 12, 2019 6:10 PM

Description

Link: https://leetcode.com/problems/power-of-four/

Tag: , **Bit manipulation**

Solution

- Approach1:

Intuition

Code

Complexity

Time: O(n)
Space: O(1)

Key: 有别的做法吗，比特操作的规律

- Approach2: iteration

Intuition

Code

```
def isPowerOfFour(self, num: int) -> bool:
    if num == 0:
        return False
    while num % 4 == 0:
        num /= 4
    return num == 1
```

Complexity

Time: O(log4(n))
Space: O(1)

Key:

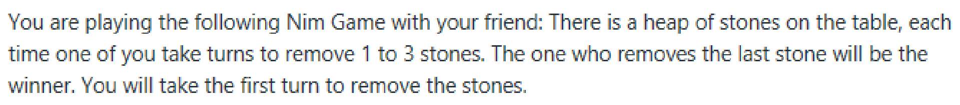
- My approach

% 292. Nim Game

Friday, July 12, 2019 6:38 PM

Description

Link: https://leetcode.com/problems/nim-game/

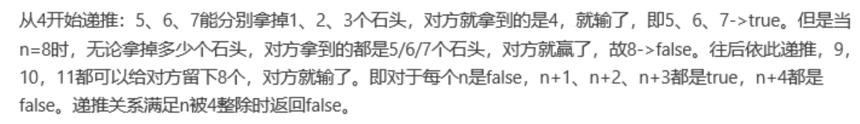

You are playing the following Nim Game with your friend: There is a heap of stones on the table, each time one of you take turns to remove 1 to 3 stones. The one who removes the last stone will be the winner. You will take the first turn to remove the stones.

Tag: ,

Solution

- Approach1:

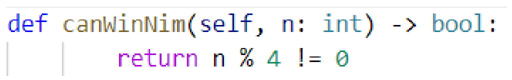
Intuition


从4开始递推：5、6、7能分别拿掉1、2、3个石头，对方就拿到的是4，就输了，即5、6、7->true。但是当n=8时，无论拿掉多少个石头，对方拿到的都是5/6/7个石头，对方就赢了，故8->false。往后依此递推，9，10，11都可以给对方留下8个，对方就输了。即对于每个n是false，n+1、n+2、n+3都是true，n+4都是false。递推关系满足n被4整除时返回false。

Complexity

Time: O(1)
Space: O(1)

Code


```
def canWinNim(self, n: int) -> bool:
        return n % 4 != 0
```

Key:

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

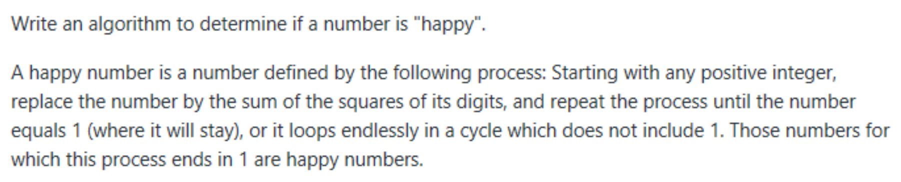
- My approach

%202. Happy Number

Friday, July 12, 2019 7:00 PM

Description

Link: https://leetcode.com/problems/happy-number/



Write an algorithm to determine if a number is "happy".

A happy number is a number defined by the following process: Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1. Those numbers for which this process ends in 1 are happy numbers.

Tag: ,

Solution

- Approach1:

Intuition

用set存出现过的数，如果产生一个出现过的数，说明陷入循环-->False

如果遇到1-->True

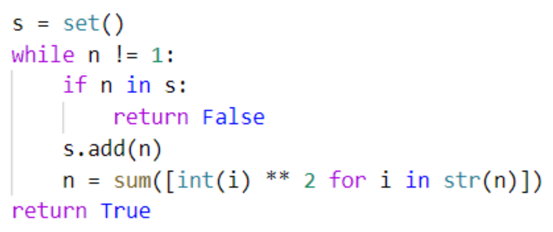
Complexity

Time: O(?)

Space: O(?)

Key: endless in a loop--> stack

Code



```
s = set()
while n != 1:
    if n in s:
        return False
    s.add(n)
    n = sum([int(i) ** 2 for i in str(n)])
return True
```

- Approach2:

Intuition

Complexity

Time: O(n)

Space: O(1)

Code

Key:

- My approach

400. Nth Digit

Friday, July 12, 2019 8:11 PM

➢ Description

Link: https://leetcode.com/problems/nth-digit/

Find the n^{th} digit of the infinite integer sequence 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ...

➢ Tag: ,

➢ Solution

- Approach1:

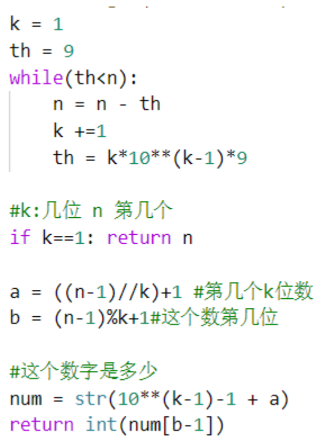
Intuition

1. It belongs to k-digit integer group, #a integer, #b digit
2. Find the exact number
3. Str(num)[b]

Complexity

Time: O(?)
Space: O(1)

Code



```
k = 1
th = 9
while(th<n):
    n = n - th
    k +=1
    th = k*10**(k-1)*9

#k:几位 n 第几个
if k==1: return n

a = ((n-1)//k)+1 #第几个k位数
b = (n-1)%k+1#这个数第几位

#这个数字是多少
num = str(10**(k-1)-1 + a)
return int(num[b-1])
```

Key:

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach

263. Ugly Number

Friday, July 12, 2019 11:21 PM

Description

Link: https://leetcode.com/problems/ugly-number/

Write a program to check whether a given number is an ugly number.

Ugly numbers are **positive numbers** whose prime factors only include `2, 3, 5`.

Tag: ,

Solution

- Approach1:

Intuition

Complexity

Time: O(?)
Space: O(1)

Code

```
if num<1: return False
while(num!=1):
    if num%2==0:
        num /= 2
    elif num%3==0:
        num /=3
    elif num%5==0:
        num /=5
    else:
        return False
return True
```

Key:

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

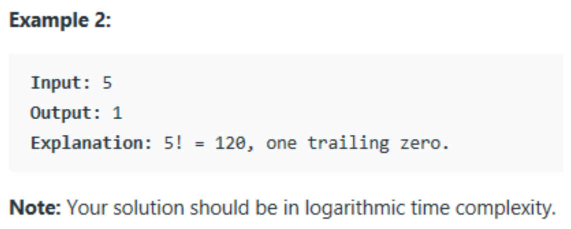
- My approach

172. Factorial Trailing Zeroes

Friday, July 12, 2019 11:45 PM

Description

Link: https://leetcode.com/problems/factorial-trailing-zeroes/



Example 2:

```
Input: 5
Output: 1
Explanation: 5! = 120, one trailing zero.
```

Note: Your solution should be in logarithmic time complexity.

Tag: ,

Solution

- Approach1:

Intuition

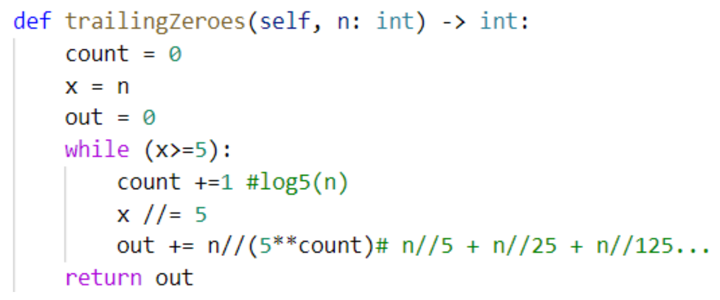
Determined by number of 5 after factorization.
Count = log_5(n)
Result = n//5 + n//5^2 +...n//5^count

Complexity

Time: O(logn)
Space: O(1)

Code

Key:



```
def trailingZeroes(self, n: int) -> int:
    count = 0
    x = n
    out = 0
    while (x>=5):
        count +=1 #log5(n)
        x //= 5
        out += n//(5**count)# n//5 + n//25 + n//125...
    return out
```

- Approach2:

Intuition

Complexity

Time: O(n)
Space: O(1)

Code

Key:

- My approach