# yifan-1 yifan-1



竞赛名称:

LeetCamp 第二期第一周竞

开始时间:

2021-12-04 04:00

登录 IP 地址:

115.216.107.249(中国杭州市电信)

邮箱:

372104922@qq.com

罚时次数:

2

结束时间:

2021-12-04 16:23

登录设备:

Windows 10 | Chrome 93.0.4577

### 答题详情

1000552



分数 0.00/20

# 下标之和

给定一个整数数组 nums ,对其从左到右扫描,并返回一个数组,数组中的第 i 个元素为 nums 中第 i 个整数前 5 次出现的下标之和,如果该整数 在此之前出现次数不满 5 次则为 -1。

#### 示例 1:

#### 提示:

```
    6 <= nums.length <= 10<sup>5</sup>
    -10<sup>4</sup> <= nums[i] <= 10<sup>4</sup>
```

用时 00:58:49

python3

提交次数 4 2021/12/04 15:17

提交结果:通过测试用例:超出时间限制42/46

语言:

执行用时: 消耗内存:

N/A

N/A

```
class Solution:
    def solve(self, nums: List[int]) -> List[int]:
        dic = {}
        res = []
        for i in range(0, len(nums)):
            if nums[i] not in dic:
                 dic[nums[i]] = 1
        else:
                 dic[nums[i]] += 1
        if nums[i] not in dic or dic[nums[i]] <= 5:
                res.append(-1)
        else:
                 s = 0</pre>
```

```
count = 0
        for j in range(i-1, -1, -1):
            if nums[j] == nums[i] and count < 5:</pre>
                s += j
                 count += 1
            if count == 5:
                 break
        res.append(s)
return res
```

#2 编程题 1000553 🕟



分数 20.00/20

# 斐波那契链表

给定一个链表的头结点 head ,依次取出其第 1, 2, 3, 5, 8, 13, 21...个元素组成新链表(若大于链表实际长度则不需要取出),并返回新链表的头结点。

#### 示例 1:

输入: head = [1,4,5,8,2]

输出: [1,4,5,2]

解释:

取出原链表中的第1,2,3,5个元素组成新链表。

## 提示:

- 链表中节点的数目在 [0, 10^6] 范围内。
- 1 <= Node.val <= 10<sup>6</sup>

用时 00:43:44

提交次数 2 2021/12/04 14:45

提交结果:

通过测试用例:

语言:

诵讨

33/33

python3

执行用时:

消耗内存:

罚时次数

1768 ms

223.8 MB

1

```
# Definition for singly-linked list.
# class ListNode:
      def __init__(self, x):
          self.val = x
          self.next = None
class Solution:
   def solve(self, head: ListNode) -> ListNode:
        if head == None:
            return None
        nums = [0] * 10000
        nums[0], nums[1] = 1, 2
        numset = set()
        numset.add(nums[0])
        numset.add(nums[1])
        for i in range(2, 10000):
            nums[i] = nums[i-2] + nums[i-1]
            if nums[i] <= 10**6:</pre>
                numset.add(nums[i])
            else:
                break
        count = 1
        newlist = ListNode(head.val)
        cur = newlist
        while head.next != None:
            head = head.next
            count += 1
            if count in numset:
                cur.next = ListNode(head.val)
                cur = cur.next
        return newlist
```

#3 编程题 1000554 🕞

分数 20.00/20

提交次数 2 2021/12/04 15:08

1

# 字符串重排

给定一个仅有大小写字符构成的字符串。,请返回一个满足以下要求的新字符串。

- 新字符串中小写字符均在左边,大写字符均在右边。
- 大写字符间的相对顺序不变, 小写字符间的相对顺序不变。

#### 示例 1:

```
输入: s = "kyClMl"
输出: "kyllCM"
```

# 提示:

- 1 <= s.length <= 10<sup>5</sup>
- s[i] 仅为大小写字母。

用时 00:22:51

提交结果: 通过测试用例: 语言:

通过 46/46 python3

执行用时: 消耗内存: 罚时次数

480 ms 16.6 MB

```
class Solution:
    def solve(self, s: str) -> str:
        n = len(s)
        res = ""
        count = 0
        for ch in s:
```

```
if ord(ch) > 90:
        res += ch
        count += 1
for ch in s:
    if ord(ch) <= 90:
        res += ch
        count += 1
return res
```

#4 编程题 1000555 🕞



分数 0.00/15

# 统计目标子序列

给定一个仅由 "a", "b", "c" 组成的字符串 tmp ,规定按 "a", "b", "c", "b", "a" 顺序依次出现的**子序列**叫做「目标子序列」,现在需要求 字符串 tmp 中「目标子序列」的个数。

注意: 答案需要以 1e9 + 7 (1000000007) 为底取模, 如: 计算初始结果为: 1000000008 , 请返回 1 。

### 示例 1:

输入: tmp = "abccaba" 输出: 2

#### 提示:

- 1 <= tmp.length <= 10<sup>5</sup>
- tmp 为仅由 "a", "b", "c" 组成的字符串。

用时 00:04:51

提交次数 0

#5 编程题 1000556 🕟



分数 0.00/15

#### 序列重排

给定长度为偶数的整数数组 nums , 再对 nums 进行 m 轮如下操作:

- 依次将 nums 中的每个元素交替插入栈和队列中,下标为偶数的插入栈中,反之插入队列中(数组下标从 0 开始)。
- 将队列中所有数出队,组成一个序列。再将栈中所有数出栈,接在之前的序列后面,这样得到长度与 nums 相同的新序列,并用其更新 nums 。

求 m 轮操作以后 nums 的第 kth 个数。

#### 示例 1:

```
输入: nums = [1,4,5,8], m = 1, kth = 4
输出: 1
解释:
1 轮操作后 nums 变为 [4,8,5,1], 第 4 个数为 1。
```

#### 提示:

- 2 <= nums.length <= 10<sup>6</sup>, 且nums.length为偶数
- 1 <= nums[i] <=10^6
- 1 <= m <= 10^6
- 1 <= kth <= nums.length

n = len(nums) // 2times= 2 \*\* n - 1

```
m %= times
if m == 0:
   m = times
for in range(m):
    res = []
    stack =[]
    queue = collections.deque()
    for i in range(0, len(nums)):
        if i % 2 == 0:
            stack.append(nums[i])
        else:
            queue.append(nums[i])
    while queue:
       res.append(queue.popleft())
    while stack:
        res.append(stack.pop())
    nums = res
return nums[kth-1]
```

编程题 1000551 🕞



分数 0.00/10

# 线性结构组合

给定的二维整数数组 types , types[i][0] = 0 表示第 i 个数据结构为栈, types[i][0] = 1 表示第 i 个数据结构为队列, types[i] [1] 表示第 i 个数据结构的大小。

再给定数组 nums , 并依次将 nums 中的每个元素插入到第 1 个数据结构中。

• 如果第 i 个数据结构已满,则第 i 个数据结构出队(出栈)一个元素并将该元素插入至第 i+1 个数据结构中。(若 i+1 > types.length,则 将该元素「输出」),再将本来要插入至第 i 个数据结构的元素插入至第 i 个数据结构中。

请返回第 kth 个「输出」的数。

#### 示例 1:

#### 输入:

types = [[0,1],[1,1]], nums = [1,4,5,8,2], kth = 2

输出: 4

#### 解释:

- 1 插入第一个数据结构「栈」中。
- 4 要插入第一个数据结构「栈」中,但此时栈满,故将 1 出栈,再将 1 插入第二个数据结构「队列」中,最后将 4 插入第一个数据结构 「栈」中。
- 5 要插入第一个数据结构「栈」中,但此时栈满,所以 4 出栈,将 4 插入第二个数据结构「队列」中,但此时队列也已满,故将 1 出队并输出,再将 4 插入第二个数结构「队列」中,最后将 5 插入第一个数据结构「栈」中。
- 8 要插入第一个数据结构「栈」中,但此时栈满,所以 5 出栈,将 5 插入第二个数据结构「队列」中,但此时队列也已满,故将 4 出队并输出,4 是第2个输出的元素,故将其返回,程序运行结束。

#### 提示:

- 1 <= types.length <= 500
- 0 <= types[i][0] <= 1
- 1 <= types[i][1] <= 3000
- 1 <= nums.length <=5000
- 1 <= kth <= nums.length sum(types[i][1])

用时 00:00:06

提交次数 0