

Assignment #9: 密集期中考试周

Updated 2130 GMT+8 Nov 6, 2023

2023 fall, Compiled by 钟明衡 物理学院

说明:

- 1) 请把每个题目解题思路 (可选), 源码Python, 或者C++ (已经在Codeforces/Openjudge上AC), 截图 (包含Accepted, 学号), 填写到下面作业模版中 (推荐使用 typora <https://typoraio.cn>, 或者用 word)。AC 或者没有AC, 都请标上每个题目大致花费时间。
- 2) 提交时候先提交pdf文件, 再把md或者doc文件上传到右侧“作业评论”。Canvas需要有同学清晰头像、提交文件有pdf、作业评论有md或者doc。
- 3) 如果不能在截止前提交作业, 请写明原因。

编程环境

操作系统: Windows_NT x64 10.0.19045

Python编程环境: Visual Studio Code 1.76.1

C/C++编程环境: Visual Studio Code 1.76.1

1. 必做题目

OJ19943: 图的拉普拉斯矩阵

matrix, <http://cs101.openjudge.cn/practice/19943/>

思路:

先创建一个 $n \times n$ 的全为0的矩阵 M

输入一组 (x, y) , 则将 $M[x][y]$ 、 $M[y][x]$ 改为 -1 , 同时将 $M[x][x]$ 和 $M[y][y]$ 加1

最后把 M 输出即可

代码

```
1 n, m = map(int, input().split())
2 M = [[0]*n for i in range(n)]
3 for i in range(m):
4     x, y = map(int, input().split())
5     M[y][x] = -1
```

```

6     M[x][y] = -1
7     M[x][x] += 1
8     M[y][y] += 1
9     for i in range(n):
10        for j in range(n):
11            print(M[i][j], end='')
12            if j != n-1:
13                print(' ', end='')
14        print('')
15

```

代码运行截图

#42300045提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

n, m = map(int, input().split())
M = [[0]*n for i in range(n)]
for i in range(m):
    x, y = map(int, input().split())
    M[y][x] = -1
    M[x][y] = -1
    M[x][x] += 1
    M[y][y] += 1
for i in range(n):
    for j in range(n):
        print(M[i][j], end='')
        if j != n-1:
            print(' ', end='')
    print('')

```

基本信息

#: 42300045

题目: 19943

提交人: 23n2300011505(12号娱乐选手)

内存: 3624kB

时间: 26ms

语言: Python3

提交时间: 2023-11-06 21:45:24

OJ19942: 二维矩阵上的卷积运算v0.2

matrix, <http://cs101.openjudge.cn/practice/19942/>

思路:

代码

```

1 m, n, p, q = map(int, input().split())
2 M = []
3 K = []
4 for i in range(m):
5     M.append(list(map(int, input().split())))
6 for i in range(p):
7     K.append(list(map(int, input().split())))
8 for i in range(m-p+1):
9     for j in range(n-q+1):
10        ans = 0

```

```

11         for y in range(p):
12             for x in range(q):
13                 ans += M[i+y][j+x]*K[y][x]
14             print(ans, end='')
15             if j != n-q+1:
16                 print(' ', end='')
17         print('')
18

```

代码运行截图

#42300266提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

m, n, p, q = map(int, input().split())
M = []
K = []
for i in range(m):
    M.append(list(map(int, input().split())))
for i in range(p):
    K.append(list(map(int, input().split())))
for i in range(m-p+1):
    for j in range(n-q+1):
        ans = 0
        for y in range(p):
            for x in range(q):
                ans += M[i+y][j+x]*K[y][x]
        print(ans, end='')
        if j != n-q+1:
            print(' ', end='')
    print('')

```

基本信息

#: 42300266

题目: 19942

提交人: 23n2300011505(12号娱乐选手)

内存: 3668kB

时间: 31ms

语言: Python3

提交时间: 2023-11-06 21:56:34

CF313B: Ilya and Queries

dp/implementation, 1100, <https://codeforces.com/contest/313/problem/B>

思路:

如果直接存每个位置是不是满足要求, 最后去数, 会超时

因此, 把从最前面到每个位置为止的数量记录在列表中, 之后直接调用第 l 、 r 个之差, 就是结果

代码

```

1 s = input()+'a'
2 L = [0]*(len(s))
3 for i in range(len(s)-1):
4     L[i+1] += L[i]
5     if s[i] == s[i+1]:
6         L[i+1] += 1
7 n = int(input())
8 for i in range(n):
9     l, r = map(int, input().split())
10    print(L[r-1]-L[l-1])
11

```

代码运行截图

General										
#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
231598052	Practice: MinghengZhong	313B - 24	Python 3	Accepted	1622 ms	4028 KB	2023-11-06 17:22:13	2023-11-06 17:22:13	★	<button>Compare</button>

→ Source	<button>Copy</button>
----------	-----------------------

```

s = input()+'a'
L = [0]*(len(s))
for i in range(len(s)-1):
    L[i+1] += L[i]
    if s[i] == s[i+1]:
        L[i+1] += 1
n = int(input())
for i in range(n):
    l, r = map(int, input().split())
    print(L[r-1]-L[l-1])

```

CF706B: Interesting drink

binary search/dp/implementation, 1100, <https://codeforces.com/problemset/problem/706/B>

思路：

将价格从小到大排序，然后使用二分查找

代码

```

1 def search(n, i, j):
2     global l
3     if j-i <= 1:
4         return i
5     else:
6         k = (i+j)//2
7         if l[k] <= n:
8             return search(n, k, j)
9         else:
10            return search(n, i, k)
11
12
13 n = int(input())
14 l = sorted([0]+list(map(int, input().split())))

```

```

15 q = int(input())
16 for i in range(q):
17     m = int(input())
18     if m >= l[-1]:
19         print(n)
20     elif m < l[1]:
21         print(0)
22     else:
23         print(search(m, 0, n))
24

```

代码运行截图

General										
#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
231603830	Practice: MinghengZhong	706B - 75	Python 3	Accepted	1060 ms	13316 KB	2023-11-06 17:40:26	2023-11-06 17:40:26	★	<button>Compare</button>

→ Source										
<pre> def search(n, i, j): global l if j-i <= 1: return i else: k = (i+j)//2 if l[k] <= n: return search(n, k, j) else: return search(n, i, k) n = int(input()) l = sorted([0]+list(map(int, input().split()))) q = int(input()) for i in range(q): m = int(input()) if m >= l[-1]: print(n) elif m < l[1]: print(0) else: print(search(m, 0, n)) </pre>										<button>Copy</button>

2. 选做题目

如果耗时太长，直接看解题思路，或者源码

CF466C: Number of Ways

binary search/brute force/data structures/dp/two pointers, 1700

<https://codeforces.com/problemset/problem/466/C>

思路：

首先，如果总和不是三的倍数，直接输出0

否则，从左右两边分别寻找可以让和为总和和 $\frac{1}{3}$ 的位置，输出所有左位置对应的可取右位置个数之和，可取的条件是右比左至少大2

查找右位置时采用二分法，防止超时

代码

```
1 def search(n, i, j):
2     global right
3     if j-i == 1:
4         return j
5     k = (i+j)//2
6     if right[k] <= n+1:
7         return search(n, k, j)
8     else:
9         return search(n, i, k)
10
11
12 n = int(input())
13 l = list(map(int, input().split()))
14 sum1 = sum(l)
15 if sum1 % 3 != 0:
16     print(0)
17 else:
18     sum1 = sum1//3
19     count = 0
20     left = []
21     right = []
22     for i in range(n):
23         count += l[i]
24         if count == sum1:
25             left.append(i)
26         if count == 2*sum1 and i != n-1:
27             right.append(i+1)
28     ans = 0
29     last = 0
30     for a in left:
31         if a >= right[-1]-1:
32             break
33         elif a <= right[last]-2:
34             ans += len(right)-last
35         else:
36             last = search(a, last, len(right)-1)
37             ans += len(right)-last
38     print(ans)
39
```

代码运行截图

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
231612669	Practice: MinghengZhong	466C - 30	Python 3	Accepted	467 ms	61796 KB	2023-11-06 18:50:36	2023-11-06 18:50:37	★	<button>Compare</button>

→ **Source**
Copy

```

def search(n, i, j):
    global right
    if j-i == 1:
        return j
    k = (i+j)//2
    if right[k] <= n+1:
        return search(n, k, j)
    else:
        return search(n, i, k)

n = int(input())
l = list(map(int, input().split()))
suml = sum(l)
if suml % 3 != 0:
    print(0)
else:
    suml = suml//3
    count = 0
    left = []
    right = []
    for i in range(n):
        count += l[i]
        if count == suml:
            left.append(i)
            if count == 2*suml and i != n-1:
                right.append(i+1)
    ans = 0
    last = 0
    for a in left:
        if a >= right[-1]-1:
            break
        elif a <= right[-1]-2:
            ans += len(right)-last
        else:
            last = search(a, last, len(right)-1)
            ans += len(right)-last
    print(ans)

```

CF1443C: The Delivery Dilemma

binary search/greedy/sortings, 1400,

<https://codeforces.com/problemset/problem/1443/C>

提示： 1) 结果要一起输出，不要分次print，会超时。 2) 用zip函数。

思路：

对于每组数据，结果为 \max (送餐时间最大值, 取餐时间总和)，只要求出这个的最小值即可

用zip可以把list绑定后排序，放在前面的优先排序

本题中，先比较送餐时间最短值和取餐时间总和，如果后者更小则直接为答案

否则，从最大的送餐时间开始，将其替换成自己取餐，这样总时间是只会减小的，一旦出现取餐总时间超过送餐最大时间且导致总时间增大，则上一个时间就是最小值

注意不能分次输出，要存到一个表里最后输出，要不然会超时

代码

```

1 t = int(input())
2 output = []
3 for _ in range(t):
4     n = int(input())
5     a = list(map(int, input().split()))
6     b = list(map(int, input().split()))
7     if sum(b) <= min(a):
8         output.append(sum(b))
9     else:

```

```

10     a, b = zip(*sorted(zip(a, b), reverse=True))
11     ans = a[0]
12     count = 0
13     for i in range(n-1):
14         count += b[i]
15         new_ans = max(count, a[i+1])
16         if new_ans < ans:
17             ans = new_ans
18         elif new_ans > ans:
19             break
20     output.append(ans)
21 for i in range(t):
22     print(output[i])
23

```

代码运行截图

General

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
231663541	Practice: MinghengZhong	1443C - 17	Python 3	Accepted	1091 ms	48012 KB	2023-11-07 06:48:55	2023-11-07 06:48:55	★	Compare

→ Source
Copy

```

t = int(input())
output = []
for _ in range(t):
    n = int(input())
    a = list(map(int, input().split()))
    b = list(map(int, input().split()))
    if sum(b) <= min(a):
        output.append(sum(b))
    else:
        a, b = zip(*sorted(zip(a, b), reverse=True))
        ans = a[0]
        count = 0
        for i in range(n-1):
            count += b[i]
            new_ans = max(count, a[i+1])
            if new_ans < ans:
                ans = new_ans
            elif new_ans > ans:
                break
        output.append(ans)
for i in range(t):
    print(output[i])

```

3. 学习总结和收获

第一次学会了用zip（之前一直是用字典来实现相同的操作的）。用zip排序会考虑每个组的排序，优先考虑放在前面的组，如果相同则按照后面的组排序。另外，注意到列表、元组都可以用zip，但zip*后的结果是元组。

另外，分次print会比存储以后单次print要慢，无特殊情况应该采用后者。