

Assignment #6: 贪心和矩阵

钟明衡2300011505

2023/10/18

说明:

- 1) 请把每个题目解题思路(可选), 源码Python, 或者C++/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。
- 3) 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。
- 4) 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

508A. Pasha and Pixels

brute force, 1100, <http://codeforces.com/problemset/problem/508/A>

Pasha loves his phone and also putting his hair up... But the hair is now irrelevant.

Pasha has installed a new game to his phone. The goal of the game is following. There is a rectangular field consisting of n row with m pixels in each row. Initially, all the pixels are colored white. In one move, Pasha can choose any pixel and color it black. In particular, he can choose the pixel that is already black, then after the boy's move the pixel does not change, that is, it remains black. Pasha loses the game when a 2×2 square consisting of black pixels is formed.

Pasha has made a plan of k moves, according to which he will paint pixels. Each turn in his plan is represented as a pair of numbers i and j , denoting respectively the row and the column of the pixel to be colored on the current move.

Determine whether Pasha loses if he acts in accordance with his plan, and if he does, on what move the 2×2 square consisting of black pixels is formed.

Input

The first line of the input contains three integers n, m, k ($1 \leq n, m \leq 1000, 1 \leq k \leq 10^5$) — the number of rows, the number of columns and the number of moves that Pasha is going to perform.

The next k lines contain Pasha's moves in the order he makes them. Each line contains two integers i and j ($1 \leq i \leq n, 1 \leq j \leq m$), representing the row number and column number of the pixel that was painted during a move.

Output

If Pasha loses, print the number of the move when the 2×2 square consisting of black pixels is formed.

If Pasha doesn't lose, that is, no 2×2 square consisting of black pixels is formed during the given k moves, print 0.

Examples

input

```
1 | 2 2 4
2 | 1 1
3 | 1 2
4 | 2 1
5 | 2 2
```

output

```
1 | 4
```

input

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

output

```
1 | 5
```

input

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

output

```
1 0
```

思路：

创建一个 $(n + 2) \times (m + 2)$ 的全为'False'的矩阵，然后模拟过程即可

代码

```
1 M = []
2
3
4 def check(i, j):
5     global M
6     if M[i+1][j] and M[i][j+1] and M[i+1][j+1]:
7         return True
8     elif M[i-1][j] and M[i][j+1] and M[i-1][j+1]:
9         return True
10    elif M[i-1][j] and M[i][j-1] and M[i-1][j-1]:
11        return True
12    elif M[i+1][j] and M[i][j-1] and M[i+1][j-1]:
13        return True
14    else:
15        return False
16
17
18 ans = 0
19 n, m, k = map(int, input().split())
20 for i in range(n+2):
21     M.append([False]*(m+2))
22 for _ in range(k):
23     x, y = map(int, input().split())
24     if ans != 0:
25         continue
26     elif check(x, y):
27         ans = _+1
28     else:
29         M[x][y] = True
```

```
30 print(ans)
31
```

代码运行截图

General

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
228570355	Practice: MinghengZhong	508A - 32	Python 3	Accepted	233 ms	8248 KB	2023-10-17 18:25:01	2023-10-17 18:25:01	★	<button>Compare</button>

→ Source

Copy

```
M = []

def check(i, j):
    global M
    if M[i+1][j] and M[i][j+1] and M[i+1][j+1]:
        return True
    elif M[i-1][j] and M[i][j+1] and M[i-1][j+1]:
        return True
    elif M[i-1][j] and M[i][j-1] and M[i-1][j-1]:
        return True
    elif M[i+1][j] and M[i][j-1] and M[i+1][j-1]:
        return True
    else:
        return False

ans = 0
n, m, k = map(int, input().split())
for i in range(n+2):
    M.append([False]*(m+2))
for _ in range(k):
    x, y = map(int, input().split())
    if ans != 0:
        continue
    elif check(x, y):
        ans = _+1
    else:
        M[x][y] = True
print(ans)
```

23555: 节省存储的矩阵乘法

implementation, matrices, <http://cs101.openjudge.cn/practice/23555/>

由于矩阵存储非常耗费空间，一个长度 n 宽度 m 的矩阵需要花费 $n*m$ 的存储，因此我们选择用另一种节省空间的方法表示矩阵。一个矩阵 X 可以表示为三元组的序列，每个三元组代表（行号，列号，元素值），如果元素值是0则我们不存储这个三元组，这样对于0很多的大型矩阵，我们节省了很多存储空间。现在我们有俩个用这种方式表示的矩阵 X 和 Y ，我们想要计算这两个矩阵的乘积，并且也用三元组形式表达，该如何完成呢。

如果不知道矩阵如何相乘，可以参考：<http://cs101.openjudge.cn/practice/18161>

输入

输入第一行是三个整数 n, m_1, m_2 ，两个矩阵 X, Y 的维度都是 $n*n$ ， m_1 是矩阵 X 中的非0元素数， m_2 是矩阵 Y 中的非0元素数。

之后是 m_1 行，每行是一个三元组（行号，列号，元素值），代表 X 矩阵的元素值，注意行列编号都从0开始。

之后是 m_2 行，每行是一个三元组（行号，列号，元素值），代表 Y 矩阵的元素值，注意行列编号都从0开始。

输出

输出是 m_3 行，代表 X 和 Y 两个矩阵乘积中的非0元素的数目，按照先行号后列号的方式递增排序。每行仍然是前述的三元组形式。

样例输入

```
1 Sample Input1:
2 3 3 2
3 0 0 1
4 1 0 -1
5 1 2 3
6 0 0 7
7 2 2 1
8
9 Sample Output1:
10 0 0 7
11 1 0 -7
12 1 2 3
13
14 解释:
15 A = [
16     [ 1, 0, 0],
17     [-1, 0, 3],
18     [0, 0, 0]
19 ]
20
21 B = [
22     [ 7, 0, 0 ],
23     [ 0, 0, 0 ],
24     [ 0, 0, 1 ]
25 ]
26
27 A*B = [
28 [7,0,0],
29 [-7,0,3],
30 [0,0,0]]
```

样例输出

```
1 Sample Input2:
2 2 2 4
3 0 0 1
4 1 1 1
5 0 0 2
6 0 1 3
7 1 0 4
8 1 1 5
9
10 Sample Output2:
11 0 0 2
12 0 1 3
13 1 0 4
14 1 1 5
15
16 解释:
17 A = [
18 [1,0],
```

```

19 [0,1]
20 ]
21
22 B = [
23 [2,3],
24 [4,5]
25 ]
26
27 A*B = [
28 [2,3],
29 [4,5]
30 ]

```

提示: tags: implementation,matrices

来源: 2021fall-cs101, hy

思路:

若 $\mathbf{C} = \mathbf{A} \times \mathbf{B}$, 则矩阵元 $c_{ij} = \sum_k a_{ik}b_{kj}$, 因此只要找到所有 k 相等的矩阵元对, 即可计算出 \mathbf{C}

代码

```

1  n, n1, n2 = map(int, input().split())
2  ans = []
3  for i in range(n):
4      ans.append([0]*n)
5  m1x = []
6  m1y = []
7  m1 = []
8  m2x = []
9  m2y = []
10 m2 = []
11 for i in range(n1):
12     x, y, num = map(int, input().split())
13     m1x.append(x)
14     m1y.append(y)
15     m1.append(num)
16 for i in range(n2):
17     x, y, num = map(int, input().split())
18     m2x.append(x)
19     m2y.append(y)
20     m2.append(num)
21 for i in range(n1):
22     for j in range(n2):
23         if m1y[i] == m2x[j]:
24             ans[m1x[i]][m2y[j]] += m1[i]*m2[j]
25 for i in range(n):
26     for j in range(n):

```

```
27         if ans[i][j] != 0:
28             print('%d %d %d' % (i, j, ans[i][j]))
29
```

代码运行截图

#41738245提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
n, n1, n2 = map(int, input().split())
ans = []
for i in range(n):
    ans.append([0]*n)
m1x = []
m1y = []
m1 = []
m2x = []
m2y = []
m2 = []
for i in range(n1):
    x, y, num = map(int, input().split())
    m1x.append(x)
    m1y.append(y)
    m1.append(num)
for i in range(n2):
    x, y, num = map(int, input().split())
    m2x.append(x)
    m2y.append(y)
    m2.append(num)
for i in range(n1):
    for j in range(n2):
        if m1y[i] == m2y[j]:
            ans[m1x[i]][m2y[j]] += m1[i]*m2[j]
for i in range(n):
    for j in range(n):
        if ans[i][j] != 0:
            print('%d %d %d' % (i, j, ans[i][j]))
```

基本信息

#: 41738245

题目: 23555

提交人: 23n2300011505(12号娱乐选手)

内存: 3696kB

时间: 48ms

语言: Python3

提交时间: 2023-10-17 23:48:46

12560: 生存游戏

matrices, <http://cs101.openjudge.cn/practice/12560/>

有如下生存游戏的规则：

给定一个 $n*m$ ($1 \leq n, m \leq 100$) 的数组，每个元素代表一个细胞，其初始状态为活着(1)或死去(0)。

每个细胞会与其相邻的8个邻居（除数组边缘的细胞）进行交互，并遵守如下规则：

任何一个活着的细胞如果只有小于2个活着的邻居，那它就会由于人口稀少死去。

任何一个活着的细胞如果有2个或者3个活着的邻居，就可以继续活下去。

任何一个活着的细胞如果有超过3个活着的邻居，那它就会由于人口拥挤而死去。

任何一个死去的细胞如果有恰好3个活着的邻居，那它就会由于繁殖而重新变成活着的状态。

请写一个函数用来计算所给定初始状态的细胞经过一次更新后的状态是什么。

注意：所有细胞的状态必须同时更新，不能使用更新后的状态作为其他细胞的邻居状态来进行计算。

输入

第一行为n和m，而后n行，每行m个元素，用空格隔开。

输出

n行，每行m个元素，用空格隔开。

样例输入

```
1 3 4
2 0 0 1 1
3 1 1 0 0
4 1 1 0 1
```

样例输出

```
1 0 1 1 0
2 1 0 0 1
3 1 1 1 0
```

来源：cs10116 final exam

思路：

为了简化对边界的处理，创建 $(n + 2) \times (m + 2)$ 的矩阵，周围一圈都为0

然后按照规则，求每个位置周边的细胞数，等于3就输出1，等于2就输出(i,j)位置上的元素，否则输出0

代码

```
1 n, m = map(int, input().split())
2 M = [[0]*(m+2)]
3 for i in range(n):
4     s = list(map(int, input().split()))
5     M.append([0]+s+[0])
6 M.append([0]*(m+2))
7 for i in range(1, n+1):
8     for j in range(1, m+1):
9         count = M[i+1][j]+M[i][j+1]+M[i-1][j]+M[i][j-1] + \
10             M[i+1][j+1]+M[i-1][j+1]+M[i-1][j-1]+M[i+1][j-1]
11         if count == 3:
12             print(1, end='')
13         elif count == 2:
14             print(M[i][j], end='')
15         else:
16             print(0, end='')
17         if j != m:
18             print(' ', end='')
19     print('')
```


代码运行截图

#41741927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]*(m+2)]
for i in range(n):
    s = list(map(int, input().split()))
    M.append([0]+s+[0])
M.append([0]*(m+2))
for i in range(1, n+1):
    for j in range(1, m+1):
        count = M[i+1][j]+M[i][j+1]+M[i-1][j]+M[i][j-1] + \
            M[i+1][j+1]+M[i-1][j+1]+M[i-1][j-1]+M[i+1][j-1]
        if count == 3:
            print(1, end='')
        elif count == 2:
            print(M[i][j], end='')
        else:
            print(0, end='')
        if j != m:
            print(' ', end='')
    print('')
```

基本信息

#: 41741927
 题目: 12560
 提交人: 23n2300011505(12号娱乐选手)
 内存: 4088kB
 时间: 34ms
 语言: Python3
 提交时间: 2023-10-18 10:06:38

04110: 圣诞老人的礼物-Santa Clau's Gifts

greedy/dp, <http://cs101.openjudge.cn/practice/04110>

圣诞节来临了，在城市A中圣诞老人准备分发糖果，现在有多箱不同的糖果，每箱糖果有自己的价值和重量，每箱糖果都可以拆分成任意散装组合带走。圣诞老人的驯鹿最多只能承受一定重量的糖果，请问圣诞老人最多能带走多大价值的糖果。

输入

第一行由两个部分组成，分别为糖果箱数正整数 n ($1 \leq n \leq 100$)，驯鹿能承受的最大重量正整数 w ($0 < w < 10000$)，两个数用空格隔开。其余 n 行每行对应一箱糖果，由两部分组成，分别为一箱糖果的价值正整数 v 和重量正整数 w ，中间用空格隔开。

输出

输出圣诞老人能带走的糖果的最大总价值，保留1位小数。输出为一行，以换行符结束。

样例输入

```
1 4 15
2 100 4
3 412 8
4 266 7
5 591 2
```

样例输出

思路:

先将礼物按照单位重量的价值 v/w 从大到小排序（对礼物的index进行操作），采用快速排序

然后按排好的顺序计算，如果能装下就整个装下，否则将最后能装下的一份礼物按比例装入，并且break

代码

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

代码运行截图

#41742340提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
v = []
w = []

def compare(i, j):
    global v, w
    if v[i]/w[i] > v[j]/w[j]:
        return 1
    elif v[i]/w[i] == v[j]/w[j]:
        return 0
    else:
        return -1

def Sort(l):
    if len(l) <= 1:
        return l
    else:
        i = len(l)//2
        return Sort([j for j in l if compare(i, j) == -1]) + [j for j in l if compare(i, j) == 0] + Sort([j for j in l if compare(i, j) == 1])

n, W = map(int, input().split())
ans = 0
for i in range(n):
    a, b = map(int, input().split())
    v.append(a)
    w.append(b)
l = Sort([i for i in range(n)])
for i in l:
    if W >= w[i]:
        W -= w[i]
        ans += v[i]
    else:
        break
```

基本信息

#: 41742340

题目: 04110

提交人: 23n2300011505(12号娱乐选手)

内存: 3652kB

时间: 21ms

语言: Python3

提交时间: 2023-10-18 10:28:01

2. 选做题目

02659: Bomb Game （选做）

matrices, <http://cs101.openjudge.cn/practice/02659/>

思路:

创建一个 $A \times B$ 的全为True的矩阵，如果T为1，就把正方形区域以外的所有元换成False，否则，将正方形区域内的所有元换成False

最后数还剩多少个True即可

代码

```
1 a, b, k = map(int, input().split())
2 bomb = []
3 ans = 0
4 for _ in range(a):
5     bomb.append([True]*b)
6 for _ in range(k):
7     x, y, r, t = map(int, input().split())
8     x -= 1
9     y -= 1
10    r = r//2
11    if t == 1:
12        for i in range(a):
13            for j in range(b):
14                if abs(i-x) > r or abs(j-y) > r:
15                    bomb[i][j] = False
16    else:
17        for i in range(x-r, x+r+1):
18            for j in range(y-r, y+r+1):
19                if i >= 0 and j >= 0 and i < a and j < b:
20                    bomb[i][j] = False
21 for i in range(a):
22     for j in range(b):
23         if bomb[i][j]:
24             ans += 1
25 print(ans)
26
```

代码运行截图

状态: Accepted

源代码

```
a, b, k = map(int, input().split())
bomb = []
ans = 0
for _ in range(a):
    bomb.append([True]*b)
for _ in range(k):
    x, y, r, t = map(int, input().split())
    x -= 1
    y -= 1
    r = r//2
    if t == 1:
        for i in range(a):
            for j in range(b):
                if abs(i-x) > r or abs(j-y) > r:
                    bomb[i][j] = False
    else:
        for i in range(x-r, x+r+1):
            for j in range(y-r, y+r+1):
                if i >= 0 and j >= 0 and i < a and j < b:
                    bomb[i][j] = False
for i in range(a):
    for j in range(b):
        if bomb[i][j]:
            ans += 1
print(ans)
```

基本信息

#: 41743766

题目: 02659

提交人: 23n2300011505(12号娱乐选手)

内存: 3700kB

时间: 112ms

语言: Python3

提交时间: 2023-10-18 11:13:58

CF545C: Woodcutters, dp/greedy, 1500 (选做)

<https://codeforces.com/problemset/problem/545/C>

思路:

从头开始处理每棵树，优先向左倒，这样不会影响后面的树。如果不能向左倒，就尝试向右倒，如果还不行就保持立着。但由于向右倒会影响后面的树，最好尝试将树向左倒。如果上一颗树向右倒，先看能不能向左，如果不能，那就继续向右，否则直立。因此每次判断时，判断顺序均为向左 → 向右 → 直立

代码

```
1 n = int(input())
2 x = []
3 h = []
4 for i in range(n):
5     xi, hi = map(int, input().split())
6     x.append(xi)
7     h.append(hi)
8 ans = 1
9 right = False
10 for i in range(1, n):
11     if i == n-1:
12         ans += 1
13     else:
14         if right:
```

```

15         if x[i]-h[i] > x[i-1]+h[i-1]:
16             ans += 1
17             right = False
18         elif x[i]+h[i] < x[i+1]:
19             ans += 1
20             right = True
21         else:
22             right = False
23     else:
24         if x[i]-h[i] > x[i-1]:
25             ans += 1
26             right = False
27         elif x[i]+h[i] < x[i+1]:
28             ans += 1
29             right = True
30         else:
31             right = False
32 print(ans)
33

```

代码运行截图

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
228640630	Practice: MinghengZhong	545C - 14	Python 3	Accepted	249 ms	10020 KB	2023-10-18 10:25:18	2023-10-18 10:25:18	★	Compare

[→ Source](#)
Copy

```

n = int(input())
x = []
h = []
for i in range(n):
    xi, hi = map(int, input().split())
    x.append(xi)
    h.append(hi)
ans = 1
right = False
for i in range(1, n):
    if i == n-1:
        ans += 1
    else:
        if right:
            if x[i]-h[i] > x[i-1]+h[i-1]:
                ans += 1
                right = False
            elif x[i]+h[i] < x[i+1]:
                ans += 1
                right = True
            else:
                right = False
        else:
            if x[i]-h[i] > x[i-1]:
                ans += 1
                right = False
            elif x[i]+h[i] < x[i+1]:
                ans += 1
                right = True
            else:
                right = False
print(ans)

```

3. 学习总结和收获

矩阵题中，矩阵边界一直是一个麻烦，于是我想到了一个简化边界判断的方法：如果原本矩阵是 $n \times m$ 的，那就用一个 $(n + 2) \times (m + 2)$ 的矩阵来存，这样就不用担心越界了。当然，不是所有的问题都可以这样处理。

最后一题一开始被标签里的"dp"给骗了，写了一个递归来枚举所有合理情况，然后因为递归层数过多RE了，之后又利用while循环实现了相同的功能，不出意外地TLE了，改成C++还是TLE，说明算法不对。想到这一章的主题是贪心，这题应该是有一个快的逻辑方法的，后来就想到了上面所说的的思路。

另外，一开始在做第一题时，初始化矩阵写了个 `M=[[False]*m]*n`，然后发现怎么样都不对，因为这样的话，每一行都指向同一个矩阵，修改其中一个元会导致一系列的值都变化。改成了for循环赋值，没这个问题了，但是不知道有没有更快的方法。