

Assignment #A: 矩阵和动态规划

Updated 1800 GMT+8 Nov 14, 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

OJ12558: 岛屿周长

matics, <http://cs101.openjudge.cn/practice/12558/>

思路:

如果一个位置是陆地, 那么它贡献的周长就是 $(4 - \text{周围陆地数量})$, 直接遍历计算就可以了

给外面套一层保护圈会更方便

代码

```

1  n, m = map(int, input().split())
2  M = [[0]*(m+2)]
3  for i in range(n):
4      M += [[0]+list(map(int, input().split()))+[0]]
5  M += [[0]*(m+2)]
6  ans = 0
7  for i in range(1, n+1):
8      for j in range(1, m+1):
9          if M[i][j] != 0:
10             ans += 4-(M[i+1][j]+M[i-1][j]+M[i][j+1]+M[i][j-1])
11 print(ans)
12

```

代码运行截图

#42477129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):
    M += [[0]+list(map(int, input().split()))+[0]]
M += [[0]*(m+2)]
ans = 0
for i in range(1, n+1):
    for j in range(1, m+1):
        if M[i][j] != 0:
            ans += 4-(M[i+1][j]+M[i-1][j]+M[i][j+1]+M[i][j-1])
print(ans)

```

基本信息

#: 42477129

题目: 12558

提交人: 23n2300011505(12号娱乐选手)

内存: 3640kB

时间: 27ms

语言: Python3

提交时间: 2023-11-14 14:53:18

OJ02760: 数字三角形

dp, <http://cs101.openjudge.cn/practice/02760/>

思路:

暴力枚举应该会超时的, 因此采用贪心+dp的思路

每个位置上保存到达这个位置时的最大值, 下一层在计算的时候就加上上面两个位置上较大的那个

每行首位都加个0保护

递推式 $l[i][j] += \max(l[i-1][j], l[i-1][j-1])$

最后的答案是最后一行里最大的那个

代码

```
1 N = int(input())
2 l = []
3 for i in range(N):
4     l += [[0]+list(map(int, input().split()))+[0]]
5 for i in range(1, N):
6     for j in range(1, i+2):
7         l[i][j] += max(l[i-1][j], l[i-1][j-1])
8 print(max(l[-1]))
9
```

代码运行截图

#42477302提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
N = int(input())
l = []
for i in range(N):
    l += [[0]+list(map(int, input().split()))+[0]]
for i in range(1, N):
    for j in range(1, i+2):
        l[i][j] += max(l[i-1][j], l[i-1][j-1])
print(max(l[-1]))
```

基本信息

#: 42477302
题目: 02760
提交人: 23n2300011505(12号娱乐选手)
内存: 3784kB
时间: 29ms
语言: Python3
提交时间: 2023-11-14 15:02:41

OJ02773: 采药

dp, <http://cs101.openjudge.cn/practice/02773>

思路:

一开始先试了下暴力枚举, 不出意料超时了

然后改用dp思路, 把每个时间能取到的最大总价值求出来, 最后一个就是答案

要先循环草药后循环时间, 否则会WA, 因为反过来就不能保证取到的是最优解

对于每个草药的循环, 时间要从后往前, 不然可能会重复取当前的草药

递推式 $dp[j] = \max(dp[j], dp[j - time[i]] + value[i])$, 同时考虑到时间应该有 $j > time[i]$

代码

```

1 T, M = map(int, input().split())
2 time = [0]*M
3 value = [0]*M
4 dp = [0]*(T+1)
5 for i in range(M):
6     time[i], value[i] = map(int, input().split())
7 for i in range(M):
8     for j in range(T, 0, -1):
9         if j >= time[i]:
10            dp[j] = max(dp[j], dp[j-time[i]]+value[i])
11 print(dp[-1])
12

```

代码运行截图

#42479110提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

T, M = map(int, input().split())
time = [0]*M
value = [0]*M
dp = [0]*(T+1)
for i in range(M):
    time[i], value[i] = map(int, input().split())
for i in range(M):
    for j in range(T, 0, -1):
        if j >= time[i]:
            dp[j] = max(dp[j], dp[j-time[i]]+value[i])
print(dp[-1])

```

基本信息

#: 42479110

题目: 02773

提交人: 23n2300011505(12号娱乐选手)

内存: 3644kB

时间: 68ms

语言: Python3

提交时间: 2023-11-14 16:14:04

OJ18106: 螺旋矩阵

matrices, <http://cs101.openjudge.cn/practice/18106/>

这个题目技巧性较强，可以看题解记住。

思路：

控制指标 (x, y) 的变化即可

具体方法是，用数组存储每次 x 和 y 的变化

如果遇到要转向，则指标 $j = (j + 1) \% 4$ 来实现循环

代码

```

1 n = int(input())
2 M = [[0]*(n+2)]
3 for i in range(n):
4     M += [[0]+[-1]*n+[0]]

```

```

5  M += [[0]*(n+2)]
6  lx = [1, 0, -1, 0]
7  ly = [0, 1, 0, -1]
8  j = 0
9  x = 0
10 y = 1
11 for i in range(1, n**2+1):
12     if M[y+ly[j]][x+lx[j]] != -1:
13         j = (j+1) % 4
14     x += lx[j]
15     y += ly[j]
16     M[y][x] = i
17 for i in range(1, n+1):
18     print(' '.join(map(str, M[i][1:-1])))
19

```

代码运行截图

#42479819提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

n = int(input())
M = [[0]*(n+2)]
for i in range(n):
    M += [[0]+[-1]*n+[0]]
M += [[0]*(n+2)]
lx = [1, 0, -1, 0]
ly = [0, 1, 0, -1]
j = 0
x = 0
y = 1
for i in range(1, n**2+1):
    if M[y+ly[j]][x+lx[j]] != -1:
        j = (j+1) % 4
    x += lx[j]
    y += ly[j]
    M[y][x] = i
for i in range(1, n+1):
    print(' '.join(map(str, M[i][1:-1])))

```

基本信息

#: 42479819

题目: 18106

提交人: 23n2300011505(12号娱乐选手)

内存: 3660kB

时间: 31ms

语言: Python3

提交时间: 2023-11-14 16:30:50

2. 选做题目

如果耗时太长，直接看解题思路，或者源码

CF189A: Cut Ribbon

brute force/dp, 1300, <https://codeforces.com/problemset/problem/189/A>

思路：

首先把 a 、 b 、 c 中小于等于 n 的不重复地存入 l 中，对于 l 中的所有 j ， $dp[j] = 1$

然后从 l 中最小的 $+1$ 位置开始dp, 递推式 $dp[i] = \max(dp[i], dp[i-j] + 1)$, 要求 $i \geq j$ 且 $dp[i-j] \neq 0$

最后一位就是答案

代码

```
1 n, a, b, c = map(int, input().split())
2 l = []
3 if a <= n and a not in l:
4     l.append(a)
5 if b <= n and b not in l:
6     l.append(b)
7 if c <= n and c not in l:
8     l.append(c)
9 dp = [0]*(n+1)
10 for j in l:
11     dp[j] = 1
12 for i in range(min(l)+1, n+1):
13     for j in l:
14         if i >= j and dp[i-j] != 0:
15             dp[i] = max(dp[i], dp[i-j]+1)
16 print(dp[-1])
17
```

代码运行截图

General									
#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged	
232725276	Practice: MinghengZhong	189A - 39	Python 3	Accepted	46 ms	0 KB	2023-11-14 12:08:39	2023-11-14 12:08:39	Compare

[→ Source](#) [Copy](#)

```
n, a, b, c = map(int, input().split())
l = []
if a <= n and a not in l:
    l.append(a)
if b <= n and b not in l:
    l.append(b)
if c <= n and c not in l:
    l.append(c)
dp = [0]*(n+1)
for j in l:
    dp[j] = 1
for i in range(min(l)+1, n+1):
    for j in l:
        if i >= j and dp[i-j] != 0:
            dp[i] = max(dp[i], dp[i-j]+1)
print(dp[-1])
```

CF455A: Boredom

dp, 1500, <https://codeforces.com/contest/455/problem/A>

思路:

将每个数字出现了的次数存到列表 l 里面

然后从头开始dp, 递推式 $dp[i] = \max(dp[i-1], dp[i-2] + l[i] * l)$

两种情况分别对应不消除*i*和消除*i*

最后一位就是答案

代码

```
1 n = int(input())
2 nums = list(map(int, input().split()))
3 N = max(nums)
4 l = [0]*(N+1)
5 for i in nums:
6     l[i] += 1
7 dp = [0, l[1]]
8 for i in range(2, N+1):
9     dp.append(max(dp[i-1], dp[i-2]+l[i]*i))
10 print(dp[-1])
11
```

代码运行截图

General

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
232729258	Practice: MinghengZhong	455A - 35	Python 3	Accepted	124 ms	13384 KB	2023-11-14 12:45:18	2023-11-14 12:45:18	★	<button>Compare</button>

→ Source

```
n = int(input())
nums = list(map(int, input().split()))
N = max(nums)
l = [0]*(N+1)
for i in nums:
    l[i] += 1
dp = [0, l[1]]
for i in range(2, N+1):
    dp.append(max(dp[i-1], dp[i-2]+l[i]*i))
print(dp[-1])
```

Copy

3. 学习总结和收获

练习了很多道dp，现在对这个算法有了更深的理解。我的理解中，dp可以写递推式主要是因为“不关心历史”，之前怎么样不会一直影响后面的。如果会影响，应该调整循环顺序来避免。另外，递推式其实比较难想，需要进行很多训练，但是最后的代码会特别简洁。