

第三次作业内容

0 / 1

编程题

1. # 现有一个地图，由横线和竖线组成（参考围棋棋盘），且每个点（点为横线与竖线的交点）种有胡萝卜，且胡萝卜有着不同的重量。

在地图上，有一只兔子，每次行走只能沿线向右或者向上移动到临近点，并吃掉该点的胡萝卜，并计算累积的胡萝卜的重量。

试补充fact函数，计算出兔子从地图的起点(地图的左下角)到终点(地图右上角)的所需要吃的胡萝卜的最小重量的值为多少，并用return返回结果。

```
def fact(inputmap):
```

在下方填充代码，注释掉pass.

示例：参数inputmap=[[1, 3, 4],

[2, 1, 2],

[4, 3, 1]]

其中数组的下标为地图的坐标，数组的值为地图上该坐标点的胡萝卜重量

现在要求的是从兔子从map[0][0]走到map[2][2]所需要吃的胡萝卜的最小重量的值。

求得路径应为map[0][0]-> map[1][0]-> map[1][1]-> map[1][2]-> map[2][2]

最小重量的值为1+2+1+2+1=7

```
pass
```

测试示例

```
map = [[1, 3, 4], [2, 1, 2], [4, 3, 1]]
```

```
print(fact(map)) # 输出为7
```

你的答案：

```
# 计算第一行和第一列的累积胡萝卜重量
for i in range(cols):
    carrot_sum[0][i] = sum(inputmap[0][:i+1])
for i in range(1, rows):
    carrot_sum[i][0] = sum(inputmap[j][0] for j in range(i+1))

# 计算其它位置的累积胡萝卜重量
for i in range(1, rows):
    for j in range(1, cols):
        carrot_sum[i][j] = inputmap[i][j] + min(carrot_sum[i-1][j],
carrot_sum[i][j-1])

# 返回从起点到终点的最小累积胡萝卜重量
return carrot_sum[-1][-1]
```

返回