

# python 实现矩阵按对角线打印

如下所示：

Description：

将一个矩阵（二维数组）按对角线向右进行打印。（搜了一下发现好像是美团某次面试要求半小时手撕的题）

Example:

Input:

```
[
[1,2,3,4],
[5,1,2,3],
[9,5,1,2]]
```

Output:

```
[[4],
[3, 3],
[2, 2, 2],
[1, 1, 1],
[5, 5],
[9]]
```

思路：考虑每条对角线开头元素的index(i,j)。i从0开始遍历，j从col-1开始遍历，首先考虑j的变化，若j变为0，则保持不变，让i变化。在确定开头元素后，可通过设置i+1，j+1及不超过范围来向lst中对角线上添加元素。

对于以上例子，各对角线开头元素的index分别为：

```
(0,3)
(0,2)
(0,1)
(0,0)
(1,0)
(2,0)
```

其中各lst中的元素index分别为：

```
[(0,3)] [(0,2),(1,3)] [(0,1),(1,2),(2,3)] [(0,0),(1,1),(2,2)] [(1,0),(2,1)] [(2,0)]
```

代码：

```
def diagonal_right(matrix):
    if not matrix:
        return []
    row = len(matrix)
    col = len(matrix[0])
    col2 = col
    result = []
    for i in range(row):
        for j in range(col2 - 1, -1, -1): #倒序遍历
            lst = []
            i1, j1 = i, j #i1,j1用于方便同一对角线元素的添加，否则改变i,j影响开头元素的选择
            while i1 <= row - 1 and j1 <= col - 1:
                lst.append(matrix[i1][j1])
            j1 += 1
            i1 += 1
            result.append(lst)
        if i == 0 and j == 0: #当遍历完(0,0)开头的一条对角线后，让j固定为0
            col2 = 1
    return('the result is: %s'%result, 'end')
```

```
#test
print(diagonal_right([[1],[2]]))
print(diagonal_right([[1]]))
print(diagonal_right([
[1,2,3,4],
[5,1,2,3],
[9,5,1,2]]))
```

输出：

```
(' the result is: [[1], [2]]', 'end')
(' the result is: [[1]]', 'end')
(' the result is: [[4], [3, 3], [2, 2, 2], [1, 1, 1], [5, 5], [9]]', 'end')
```

变型：由(0,0)开始向左打印：

**思路：**同样考虑每条对角线开头元素的index(i,j)。i从0开始遍历，j也从0开始遍历，首先考虑j的变化，若j变为col-1，则保持不变，让i变化。在找到开头元素后，可通过设置i+1，j-1及不超过范围来向lst中对角线上添加元素。

对于以上例子，各对角线开头元素的index分别为：

(0,0)  
(0,1)  
(0,2)  
(0,3)  
(1,3)  
(2,3)

其中各lst中的元素index分别为：

[(0,0)] [(0,1),(1,0)] [(0,2),(1,1),(2,0)] [(0,3),(1,2),(2,1)] [(1,3),(2,2)] [(2,3)]

代码：

```
def diagonal_left(matrix):
    if not matrix:
        return []
    row = len(matrix)
    col = len(matrix[0])
    k=0
    result = []
    for i in range(row):
        for j in range(k,col): # j顺序遍历
            lst = []
            i1, j1 = i, j # i1,j1用于方便同一对角线元素的添加，否则改变i,影响开头元素的选择
            while i1 <= row - 1 and j1 >=0:
                lst.append(matrix[i1][j1])
                j1 -= 1
                i1 += 1
            if i==0 and j==col-1:
                k=col-1
            result.append(lst)
    return ('the result is: %s' % result, 'end')

# test
print(diagonal_left([[1],[2]]))
print(diagonal_left([[1]]))
print(diagonal_left([
[1,2,3,4],
[5,1,2,3],
[9,5,1,2] ]))
```

输出：

```
('the result is: [[1], [2]]', 'end')
('the result is: [[1]]', 'end')
('the result is: [[1], [2, 5], [3, 1, 9], [4, 2, 5], [3, 1], [2]]', 'end')
```

此问题还可以有多种变型，若从左下角，右下角按对角线打印等，只需要进行简单的修改即可实现。

以上这篇python 实现矩阵按对角线打印就是小编分享给大家的全部内容了，希望能给大家一个参考，也希望大家多多支持软件开发网。

**您可能感兴趣的文章:**python 实现一个反向单位矩阵示例python 实现矩阵填充0的例子Python操作多维数组输出和矩阵运算示例Python numpy.zeros() 初始化矩阵实例Python实现随机取一个矩阵数组的某几行python将邻接矩阵输出成图的实现