

数字三角形

1 题目描述

```
  7
 3 8
8 1 0
2 7 4 4
4 5 2 6 5
```

如上图所示，从一个数字三角形的顶部走到底部有很多条不同的路径，规则是只能从当前节点走到下一层相邻的节点，即下一层的左边或右边。例如第三行第二个数字“1”只能走到第四行的第二个数字“7”与第三个数字“4”。请寻找最佳一条路径，使得这条路径上节点的数字总和最大。

1.1 输入描述：

输入包含多组。每组数据的第一行包含一个正整数 n ($1 \leq n \leq 100$)，代表三角形的层数。紧接着有 n 行数字，第 i ($1 \leq i \leq n$) 行包含 i 个自然数。

1.2 输出描述：

对应每组数据，输出最大的和。

1.3 输入例子：

```
5
7
3 8
8 1 0
2 7 4 4
4 5 2 6 5
```

1.4 输出例子：

```
30
```

2 解题思路

将示例的输入转换成二维数组。如图 1 所示。

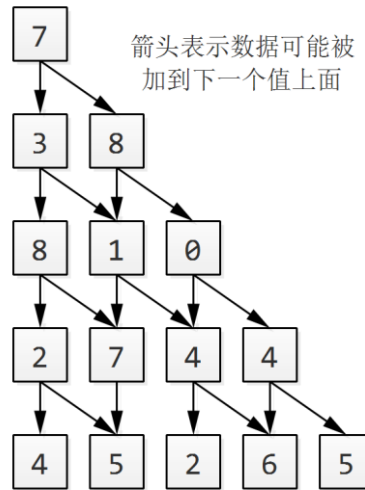


图 1 数据流动图

根据意思和图 1 分析，本题可以使用动态规划解决。设有待求最大值的矩阵 **matrix**，它有 **n** 行，第 **i** 行最多 **i** 列，每个元素的值都是正数。第 **i**（从 1 开始计算）行第 **j**（从 1 开始计算）列对应到 **matrix** 的位置为 **matrix[i-1][j-1]**。

第一列，它的第一行只有一个元素，它的最大值就是其自身，对于其它行，第一列的最大值等于它上一个元素的最大值加上当前值。对象线上的元素，除第一个元素外，其它元素的最大值等于它的左上角元素的最大值和当前值的和。最后对矩阵中的其它元素，其最大值等于左上角元素的最大值和上方元素的最大两都取较大的，再和当前值相加。对于 **matrix[i][j]** 有递推方程

$$matrix[i][j] = \begin{cases} matrix[i][j] & i = 0 \text{ and } j = 0 \\ matrix[i-1][j] + matrix[i][j] & i > 0 \text{ and } j = 0 \\ matrix[i-1][j-1] + matrix[i][j] & i = j > 0 \\ \max\{matrix[i-1][j-1], matrix[i-1][j]\} + matrix[i][j] & i > j > 0 \end{cases}$$

最后，要求出 **matrix** 中的最大值只要遍历最后一列，它们中的最大值就是所求的 **matrix** 的最大值。