

Maximum path sum I

作者: [taujiong](#)

原文链接: <https://ld246.com/article/1577803972473>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



Project Euler 第 18 题

<!-- more -->

Question

By starting at the top of the triangle below and moving to adjacent numbers on the row below, the maximum total from top to bottom is 23.

3
7 4
2 4 6
8 5 9 3

That is, $3 + 7 + 4 + 9 = 23$.

Find the maximum total from top to bottom of the triangle below:

75
95 64
17 47 82
18 35 87 10
20 04 82 47 65
19 01 23 75 03 34
88 02 77 73 07 63 67
99 65 04 28 06 16 70 92

```

41 41 26 56 83 40 80 70 33
41 48 72 33 47 32 37 16 94 29
53 71 44 65 25 43 91 52 97 51 14
70 11 33 28 77 73 17 78 39 68 17 57
91 71 52 38 17 14 91 43 58 50 27 29 48
63 66 04 68 89 53 67 30 73 16 69 87 40 31
04 62 98 27 23 09 70 98 73 93 38 53 60 04 23

```

NOTE: As there are only 16384 routes, it is possible to solve this problem by trying every route. However, Problem 67, is the same challenge with a triangle containing one-hundred rows; it cannot be solved by brute force, and requires a clever method! ;o)

Analysis

1. 第 i 行第 j 个数只能由第 $i-1$ 行第 $j-1$ 个数或者第 $i-1$ 行第 j 个数加上自身得到，除每行的首列尾列之外，应该有

$$A_{i,j} += \max \{A_{i-1,j-1}, A_{i-1,j}\}$$

2. 要求最大的和，只要求最后一行的最大值，即通过步骤 1 的思路递归往前求即可。

3. (2019.11.6) 按照自上而下的方式需要单独考虑首列和尾列，而按照自下而上的方式则会使公式更统一。

4. (2019.11.6) get_max 的 lambda 表达式纯属脱裤子放屁。

Program

```

# 自上而下
numbers = [[75],
            [95, 64],
            [17, 47, 82],
            [18, 35, 87, 10],
            [20, 04, 82, 47, 65],
            [19, 01, 23, 75, 03, 34],
            [88, 02, 77, 73, 07, 63, 67],
            [99, 65, 04, 28, 06, 16, 70, 92],
            [41, 41, 26, 56, 83, 40, 80, 70, 33],
            [41, 48, 72, 33, 47, 32, 37, 16, 94, 29],
            [53, 71, 44, 65, 25, 43, 91, 52, 97, 51, 14],
            [70, 11, 33, 28, 77, 73, 17, 78, 39, 68, 17, 57],
            [91, 71, 52, 38, 17, 14, 91, 43, 58, 50, 27, 29, 48],
            [63, 66, 04, 68, 89, 53, 67, 30, 73, 16, 69, 87, 40, 31],
            [04, 62, 98, 27, 23, 09, 70, 98, 73, 93, 38, 53, 60, 4, 23]]

```

```
get_max = lambda a, b: a if a > b else b
```

```

rows = len(numbers)
for i in range(1, rows):
    columns = len(numbers[i])
    for j in range(columns):

```

```

    a = 0 if j == 0 else numbers[i - 1][j - 1]
    b = 0 if j == len(numbers[i - 1]) else numbers[i - 1][j]
    numbers[i][j] += get_max(a, b)

rst = max(numbers[-1])
print(rst)

# 自下而上
numbers = [[75],
           [95, 64],
           [17, 47, 82],
           [18, 35, 87, 10],
           [20, 04, 82, 47, 65],
           [19, 01, 23, 75, 03, 34],
           [88, 02, 77, 73, 07, 63, 67],
           [99, 65, 04, 28, 06, 16, 70, 92],
           [41, 41, 26, 56, 83, 40, 80, 70, 33],
           [41, 48, 72, 33, 47, 32, 37, 16, 94, 29],
           [53, 71, 44, 65, 25, 43, 91, 52, 97, 51, 14],
           [70, 11, 33, 28, 77, 73, 17, 78, 39, 68, 17, 57],
           [91, 71, 52, 38, 17, 14, 91, 43, 58, 50, 27, 29, 48],
           [63, 66, 04, 68, 89, 53, 67, 30, 73, 16, 69, 87, 40, 31],
           [04, 62, 98, 27, 23, 09, 70, 98, 73, 93, 38, 53, 60, 4, 23]]

rows = len(numbers)
for i in range(rows - 2, -1, -1): # 从倒数第二行开始到第一行
    columns = len(numbers[i])
    for j in range(columns):
        a = numbers[i + 1][j]
        b = numbers[i + 1][j + 1]
        numbers[i][j] += max(a, b)

rst = numbers[0][0]
print(rst)

```