

Maximum path sum II

Problem 67 (<https://projecteuler.net/problem=67>)

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 in [triangle.txt \(p067_triangle.txt\)](#) (right click and 'Save Link/Target As...'), a 15K text file containing a triangle with one-hundred rows.

NOTE: This is a much more difficult version of Problem 18. It is not possible to try every route to solve this problem, as there are 299 altogether! If you could check one trillion (10^{12}) routes every second it would take over twenty billion years to check them all. There is an efficient algorithm to solve it. ;o)

Solution

```
In [1]: ▶ with open('p067_triangle.txt', 'r') as f:
        data = f.read()

        mx = [[int(y) for y in x.split()] for x in data.splitlines()]

        for r in range(1, len(mx)):
            for c in range(len(mx[r-1])):
                mx[r][c] += max(mx[r-1][c], (mx[r-1][c-1] if c else 0))

        print(max(mx[-1]))
```

7273