

# Problem 18 - Maximum Path Sum I

Gautam Manohar

11 June 2018

*This document originally appeared as a blog post on my website. Find it at [gautammanohar.com/euler/18](http://gautammanohar.com/euler/18).*

## 1 Problem Statement

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

```
      3
     7 4
    2 4 6
   8 5 9 3
```

**Figure 1.** Note that  $3 + 7 + 4 + 9 = 23$ .

Find the maximum path length for a given such triangle.

## 2 My Algorithm

We use a data structure that mimics the structure of a given triangle.

For each row, we update a **parent** array to store the maximal path sums that end in each entry of the row above. We process each row of the tree into an array called **current**. For each element **x** in the row, we add to current the sum of **x** and the maximum of the sums corresponding to the two (or one) entries adjacent to **x** in the previous row. These sums are found in **parent**.

When we finish processing a row, we set **parent** equal to **current** and continue until all the rows have been processed. The final answer is the maximal element in **parent**. This solution has time complexity  $O(n^2)$ , because we perform an operation for each element of the triangle.

### 3 Other Solutions

A brute force solution is possible for small triangles. The number of paths is the sum of the number of paths that end in each entry of the bottom row. This is the sum of the entries in the  $n$ -th row of Pascal's triangle, or  $2^n$ . And so a brute force solution would have time complexity  $O(2^n)$ .