

Euler 0011

The Problem:

Find the greatest product of 4 adjacent numbers in the grid (which can be found on the PE website).

This can be up, down, left, right, or diagonal.

Considerations:

- We don't need to check left ever. In the last 3 columns we don't check right, either.
- We don't need to check up ever. In the last 3 rows we don't check down.
- We don't need to check up diagonals. We will check both left and right diagonals, and ignore the relevant diagonals when we are in the first or last 3 columns. We don't need to check diagonals in the last 3 rows.

The Approach:

Starting from 0,0 being the topmost leftmost corner of the grid, we can keep iterating through and checking every case, remembering the considerations above.

The Code:

```
import numpy
from enum import Enum

class Direction(Enum): #not necessary to the solution, but helps me as a human visualize
    where the solution is at
        DOWN = "DOWN"
        RIGHT = "RIGHT"
        LEFT_DIAG = "LEFT_DIAG"
        RIGHT_DIAG = "RIGHT_DIAG"
```

```

best_index = [0,0]
best_dir = Direction.DOWN
best_product = 0
length = 4
grid = numpy.genfromtxt("numbers", delimiter=" ") #using numpy to read in the file to a
grid for us

for i in range(len(grid)): #go through every row in the grid
    for j in range(len(grid[i])): #go through every column in the grid
        #print(i,j)
        #check right
        if j < len(grid)-(length-1):
            current_prod = 1
            for k in range(length):
                current_prod *= grid[i,j+k]
            if current_prod >= best_product:
                best_index = [i,j]
                best_dir = Direction.RIGHT
                best_product = current_prod

        #check down
        if i < len(grid)-(length-1):
            current_prod = 1
            for k in range(length):
                current_prod *= grid[i+k,j]
            if current_prod >= best_product:
                best_index = [i,j]
                best_dir = Direction.DOWN
                best_product = current_prod

        #check down right
        if i < len(grid)-(length-1) and j < len(grid)-(length-1):
            current_prod = 1
            for k in range(length):
                current_prod *= grid[i+k,j+k]
            if current_prod >= best_product:
                best_index = [i,j]
                best_dir = Direction.RIGHT_DIAG

```

```
        best_product = current_prod

#check down left
if i < len(grid)-(length-1) and j > (length-1):
    current_prod = 1
    for k in range(length):
        current_prod *= grid[i+k,j-k]
    if current_prod >= best_product:
        best_index = [i,j]
        best_dir = Direction.LEFT_DIAG
        best_product = current_prod

print(best_index, best_dir, best_product) #prints out the best position, the direction it
points, and the product
```

Revision #1

Created 2025-08-21 16:58:58 UTC by Maxwell

Updated 2025-08-21 17:05:10 UTC by Maxwell