Project Euler

# Abstract

Take a problem from Project Euler and solve it with Python. With 725 problems to choose from everyone should be able to find a unique problem.

# Problem Statement

The sample problem I choose was the ***Special Pythagorean Triplet.*** The problem consisted of a few moving parts that had to be met before begin able to solve the problem.

A Pythagorean triplet is a set of three natural numbers, , for which,

For example,

There exists exactly one Pythagorean triplet for which.

Find the product of .

# Methodology

Utilizing Python we can make a series for loops to iterate through each variation until it is found with by the, if statement. The only library we need to do this is the math library. At first I utilized three for loops to get the result, but upon thinking on the problem. The variable can be calculated as,. Eliminating the third for loop allowed the same calculation but much faster. The calculation went from 18 seconds to roughly 0.05 seconds.

First Iteration of code

'''

Chris Oakley

Functions Part 1

11/13/2020

EGN3214 - Assignment 4

Variables:

a,b,c - The sides of the triangle

numbreak - The max number the your looking for squared

The end result should equal out to 31875000 as the number we are looking for.

'''

import math

numbreak = 1000

break\_for = False

for a in range(1,numbreak):

for b in range(a+1,numbreak):

for c in range(b+1,numbreak):

if (a+b+c)==numbreak and (a\*\*2+b\*\*2)==c\*\*2:

print(a\*b\*c)

break\_for = True

if break\_for == True:

break

if break\_for == True:

break

Last iteration of code

'''

Chris Oakley

Functions Part 1

11/13/2020

EGN3214 - Assignment 4

Variables:

a,b,c - The sides of the triangle

numbreak - The max number the your looking for squared

The end result should equal out to 31875000 as the number we are looking for.

'''

import math

numbreak = 1000

break\_for = False

for a in range(1,numbreak):

for b in range(a+1,numbreak):

if (a+b+c)==numbreak and (a\*\*2+b\*\*2)==c\*\*2:

print(a\*b\*c)

break\_for = True

if break\_for == True:

break

# Solution

The following variables solved the problem;

# Conclusion

This problem created some interesting though processes. By the first iteration of the code using three for loops and seeing how it took literally forever in the space of a computer. By optimizing the code for a better solution you decrease the time it takes to get to said solution. Looking over some other problems at Project Euler a thought came to mind that all the problems are looking for a optimized solution. The solution is simply the endpoint of the problem. The journey is solving it with critical thinking.