Project Euler Problem 273: Sum of Squares

Wee JunJie

January 1, 2020

1 Problem

Consider equations of the form: $a^2 + b^2 = N$, $0 \leq a \leq b$, a, b and N integer. For $N = 65$ there are two solutions: $a = 1$, $b = 8$ and $a = 4$, $b = 7$. We call $S(N)$ the sum of the values of a of all solutions of $a^2 + b^2 = N$, $0 \leq a \leq b$, a, b and N integer. Thus $S(65) = 1 + 4 = 5$. Find $\sum S(N)$, for all squarefree N only divisible by primes of the form $4k + 1$ with $4k + 1 < 150$.

2 Understanding the Difficulty of this Problem

Let (a, b) denote a pair of a and b whose sum of squares gives the value N. At first glance, one might think that the pairs (a, b) represent no special meaning in its numbers. However, we can first generate all the possible squarefree numbers N needed in this problem. For this problem, the code below should suffice to get all possible N.

```
nums = []

def GenNum(num, ind):
    """ Generates Squarefree Numbers from prime numbers of 4*k + 1 < 150"""
    for a in range(exp[ind]+1):
        if ind == len(subprimes)-1:
            nums.append(num*subprimes[ind])
        else:
            if a+1 < exp[ind]+1:
                nums.append(num*subprimes[ind])
            GenNum(num, ind+1)
            num *= subprimes[ind]

primes = list(sp.sieve.primerange(1, 150))
subprimes = []
for p in primes:
    if p%4 == 1: # Take only prime numbers of 4*k+1
        subprimes.append(p)
exp = [1 for i in range(len(subprimes))]
GenNum(1, 0)
nums = np.unique(np.sort(nums))
```

This is where it gets interesting. Note that the first few terms of N should be of the following:

$$5, 13, 17, 29, 37, 41, 53, 61, 65, 73, \cdots$$

If one were to take the first few terms of N and search in OEIS for any clues into its sequence, then the only connection/link one could get is this: The numbers N actually represents the hypotenuse of primitive Pythagorean triples whose N is also squarefree and are only divisible by primes of $4k+1 < 150$. Recall that primitive Pythagorean triples (x, y, z) represents triples where $x^2+y^2 = z^2$ and $\gcd(x, y, z) = 1$. Hence, one possible sequence OEIS will give you would be A008846 or A020882. Note that A020882 gives the hypotenuses of primitive Pythagorean triples with multiplicity. For example, for $N = 65$, we have pairs $(1, 8)$ and $(4, 7)$ which will produce two distinct primitive Pythagorean triangles. Notice that I still have not explained what that pairs (a, b) represent in this context. In fact, the pairs are generator pairs for primitive Pythagorean triples which means they are values a and b which under a set of formulas, can generate all the possible primitive Pythagorean triples (x, y, z). Mathematically, we have the following well-known set of formulas:

$$x = a^2 - b^2, y = 2ab, N = z = a^2 + b^2,$$

where $N = z$ is the hypotenuse of the primitive Pythagorean triangle. Note that the generator pairs are in fact pairs (a, b) which are always one even and one odd (e.g. (4,7) or (1,8)) and satisfy $\gcd(a, b) = 1$. Having explaining so much information, why is this problem still having a difficulty of 70%? This is because by simply coding the pairs (a, b) to generate all the possible primitive Pythagorean triples is much less difficult. This problem is difficult because it requires us to perform the inversion. i.e. Given just the hypotenuse N, generate all its possible (a, b)s. At this point, you might be tempted to find the internet for any set of inverse formulas to find (a, b) given N.

3 Solution

Due to the usefulness of the sympy library, a simple Python code can still solve the problem in finite time. :)

```
from sympy.solvers.diophantine import cornacchia

sum = 0
for n in nums:
    sum += np.sum(np.unique(np.sort(list(cornacchia(1,1,n)), axis =
    1), axis = 0)[:,0])
print("Ans: ", sum)
```

Note that there are still even more efficient solutions out there, some applying Gaussian Integers or Serret's Algorithm.