Project_Euler_021

February 4, 2018

1 Project Euler Problem 21

Let d(n) be defined as the sum of proper divisors of n (numbers less than n which divide evenly into n). If d(a) = b and d(b) = a, where a b, then a and b are an amicable pair and each of a and b are called amicable numbers.

For example, the proper divisors of 220 are 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 and 110; therefore d(220) = 284. The proper divisors of 284 are 1, 2, 4, 71 and 142; so d(284) = 220.

Evaluate the sum of all the amicable numbers under 10000.

In [5]:
```
from math import sqrt, ceil

amicable_sum = 0

# The function divisorsum(k) gets the sum of proper divisors of k
# by going through all numbers i from 2 to sqrt(k) and checking
# divisibility.  If i is the square root of k, we just add that
# to the sum of divisors.  Otherwise, if i is a divisor of k
# less than the square root of k, we add i and k/i to
# the sum of divisors.  Remember that 1 is always a divisor of
# any integer, so we add that at the end.

def divisorsum(k):
    total = 0
    for i in range(2, int(ceil(sqrt(k))) + 1):
        if k / i == i and k % i == 0:
            total += i
        elif i > sqrt(k):
            total += 0
        elif k % i == 0:
            total += i
            total += k/i
    return int(total + 1)

# A number j is part of an amicable pair if the divisor sum of its
# divisor sum is equal to j, and if the divisor sum of j isn't
# equal to j.
```

```
        for j in range(5, 10000):
            if (divisorsum(divisorsum(j)) == j and
                divisorsum(j) != j):
                amicable_sum += j

        print("The sum of amicable numbers under 10000 is {}"
              .format(amicable_sum))
```

```
The sum of amicable numbers under 10000 is 31626
```