

Project_Euler_023

February 4, 2018

1 Project Euler Problem 23

A perfect number is a number for which the sum of its proper divisors is exactly equal to the number. For example, the sum of the proper divisors of 28 would be $1 + 2 + 4 + 7 + 14 = 28$, which means that 28 is a perfect number.

A number n is called deficient if the sum of its proper divisors is less than n and it is called abundant if this sum exceeds n .

As 12 is the smallest abundant number, $1 + 2 + 3 + 4 + 6 = 16$, the smallest number that can be written as the sum of two abundant numbers is 24. By mathematical analysis, it can be shown that all integers greater than 28123 can be written as the sum of two abundant numbers. However, this upper limit cannot be reduced any further by analysis even though it is known that the greatest number that cannot be expressed as the sum of two abundant numbers is less than this limit.

Find the sum of all the positive integers which cannot be written as the sum of two abundant numbers.

```
In [10]: from math import sqrt, ceil

# We'll recycle the divisorsum() function from problem 21.

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)

# abundantList will contain all abundant numbers up to 28123.

abundantList = []
for j in range(1, 28124):
    if divisorsum(j) > j:
        abundantList.append(j)
```

```

# Now, create a list of numbers that are not the sum of
# two abundant numbers by taking them away from the set of
# all integers up to 28123.

resultSet = set(range(28124))
for k in range(len(abundantList)):
    for m in range(len(abundantList)):
        if (abundantList[k]+abundantList[m]) in resultSet:
            resultSet.remove(abundantList[k] + abundantList[m])

print("The sum is {}".format(sum(list(resultSet))))

```

The sum is 4179871.