Euler Project 70

Joshua Van Rossum

03/27/2024

Problem Statement

Euler's totient function, $\phi(n)$ [sometimes called the phi function], is used to determine the number of positive numbers less than or equal to n which are relatively prime to n. For example, as $1, 2, 4, 5, 7$, and 8, are all less than nine and relatively prime to nine, $\phi(9) = 6$.
The number 1 is considered to be relatively prime to every positive number, so $\phi(1) = 1$.

Interestingly, $\phi(87109) = 79180$, and it can be seen that 87109 is a permutation of 79180.

Find the value of n, $1 < n < 10^7$, for which $\phi(n)$ is a permutation of n and the ratio $n/\phi(n)$ produces a minimum.

Answer

8,319,823

Idea

To get $min(\frac{n}{\phi(n)})$ we want $\phi(n)$ to be as large as possible. The easiest way to get the largest $\phi(n)$ is to take the largest prime in the interval and subtract 1. However n and $\phi(n)$ must also be anagrams of each other and no n and (n-1) satisfy that condition. So we start looking at n's with two prime factors. I created a simple helper method that ensures that the n we are checking has only two prime factors. I also create a method to check if two numbers are anagrams. I have a simple equation to calculate phi = round((i*j) * (1 - (1/i)) * (1 - (1/j))), where i and j are prime factors. Since we want $\phi(n)$ to be maximum we start looking for the largest prime factors possible, which means we should start looking at primes around the square root of 10,000,000 = 3163. I created two loops to test different combinations of primes and kept track of the smallest ratio found.

Python Code

```
def anagram(n1, n2):
  return sorted(str(n1)) == sorted(str(n2))

def num_factors(n):
    factors = 0
    divisor = 2
    while n > 1:
        while n % divisor == 0:
            factors += 1
            n //= divisor
            if (factors > 2):
              return False
        divisor += 1

    return True

minimal = 10
min_phi = -1

for i in range(3163, 1, -1):
  j = i
  while i * j < 10000000:
    p = round((i*j) * (1 - (1/i)) * (1 - (1/j)))
    if ( anagram(p, (i*j)) & (((i*j) / p) < minimal) & num_factors(i*j)):
      print(i*j)
      minimal = ((j * i) / p)
      print(minimal)
      min_phi = i * j
    j += 1

print(min_phi)
```