Fall 2021 CIS 3362 Homework #5: Number Theory Solutions

Written by Zachariah Abueg

1) Without the aid of a computer program, determine the prime factorization of 808995600. Show your work. You may do division in a calculator.

Solution

Of course, there are a few ways to prime factor this number. The way I chose to do it was to keep dividing it by prime numbers until we reach 1 and creating the prime factorization from there. In fact, this is equivalent to those prime factorization trees we used to make back then.

Notice that our number, $808,995,600$, ends in two zeros, so we know that it is a multiple of 100. Now, $100 = 10^2 = (2 \times 5)^2 = 2^2 \times 5^2$, so right off the bat, we have 2^2 and 5^2 in our prime factorization. Dividing $808,995,600$ by 100, we have $8,089,956$. 808,995,600

$$808,995,600 = 2^2 \times 5^2 \times 8,089,956$$

From here, we can keep dividing by 2 until we reach a non-integer. Using the calculator, we can divide $8,089,956$ by 2 two more times to get $2,022,489$, after which dividing by 2 gives a non-integer. After having divided by 2 two times, we know have an additional 2^2 factor, giving 2 in our prime factorization the exponent $2^{2+2} = 2^4$.

$$808,995,600 = 2^4 \times 5^2 \times 2,022,489$$

Now we are at $2,022,489$ after dividing out all of the 2s from our original number. From here, let's divide $2,022,489$ by the next prime, 3, until we reach a non-integer. Using the calculator, we can divide by 3 five times to get $8,323$, after which dividing by 3 gives a non-integer. After having divided by 3 five times, we now have an additional factor of 3^5.

$$808,995,600 = 2^4 \times 3^5 \times 5^2 \times 8,323$$

Now we are at $8,323$. Clearly, dividing by 5 is futile, as we know that all multiples of 5 end in either 0 or 5 and we can see that $8,323$ does not. Thus, we have divided out all 5s from our original number. We then try dividing 169,169 by the next prime number, 7, until we reach a non-integer, which we can do one time to reach $1,189$. Thus, 7^1 is a factor.

$$808,995,600 = 2^4 \times 3^5 \times 5^2 \times 7^1 \times 1,189$$

After that, we try dividing $1,189$ by 11, which does not yield us a non-integer. We try again for 13, then for 17, then for 19,... all the way until 29, at which point we yield an integer result. We can divide by 29 once to get 41, giving us a factor of 29^1. Finally, we can divide by 41 once since 41 itself is prime, for a final factor of 41^1. Our final prime factorization is

$$808,995,600 = 2^4 \times 3^5 \times 5^2 \times 7^1 \times 29^1 \times 41^1.$$

2) What is φ(808995600)?

Solution

We use the fact that $\varphi(n)$ is multiplicative and the fact that $\varphi(p^k) = p^k - p^{k-1}$ for a prime p and positive integer k, to get

$$\varphi(808995600) = \varphi\left(2^4 \times 3^5 \times 5^2 \times 7^1 \times 29^1 \times 41^1\right) =$$
$$\varphi\left(2^4\right) \times \varphi\left(3^5\right) \times \varphi\left(5^2\right) \times \varphi\left(7^1\right) \times \varphi\left(29^1\right) \times \varphi\left(41^1\right)$$
$$\left(2^4 - 2^3\right) \times \left(3^5 - 3^4\right) \times \left(5^2 - 5^1\right) \times \left(7^1 - 7^0\right) \times \left(29^1 - 29^0\right) \times \left(41^1 - 41^0\right) =$$
$$8 \times 162 \times 20 \times 6 \times 28 \times 40 = 174,182,400.$$

3) Using Fermat's Theorem, determine the remainder when 12^{11764} is divided by 1471.

Solution

How will we use Fermat's Theorem to find the remainder of this huge power?

First note that the remainder when x is divided by y is equivalent to $x \pmod{y}$, so we want to find $12^{11764} \pmod{1471}$. Now, recall that Fermat's Theorem states that if p is prime and $\gcd(a, p) = 1$, then $a^{p-1} \equiv 1 \pmod{p}$.

We know from exponent rules that

$$(a^b)^c = a^{b \times c} \tag{1}$$

or

$$a^{b \times c} = (a^b)^c \tag{2}$$

If we have from Fermat's Theorem that $a^{p-1} \equiv 1 \pmod{p}$ for some a and p, then we can exploit this congruence to raise higher powers of a:

Say we had a number greater than $p - 1$ that is a multiple of $p - 1$, let's say $k \times (p - 1)$ for some positive integer k, and we wanted to raise a to that power, $k \times (p - 1)$, and take the result mod p. That is, we want to know $a^{k \times (p-1)} \pmod{p}$. Then we use (2) to get

$$a^{k \times (p-1)} = a^{(p-1) \times k} = \left(a^{p-1}\right)^k$$

and now, since we know from Fermat's Theorem that $a^{p-1} \equiv 1 \pmod{p}$, we have

$$\left(a^{p-1}\right)^k \equiv 1^k = 1 \pmod{p}.$$

[Aside: Why can we do this? Why can we "replace" $\left(a^{p-1}\right)^k$ with 1^k when working (mod p)? It is due to the rules of modular arithmetic. Suppose $a \equiv b \pmod{c}$. Then the rules of modular arithmetic tell us that multiplying a and b by the same integer are equivalent (mod c) - that is,

$d \times a \equiv d \times b \pmod{c}$ for any integer d. So if we wanted to find $d \times a \pmod{c}$, we could simply find $d \times b \pmod{c}$ if it's easier to calculate instead.

Now, what would happen if we raised a and b to the same positive integer? Say we raise a and b to a positive integer d. We know that a^d is the same as multiplying a by itself d times and b^d is the same as multiplying b by itself d times. Now by the above statement, we have that multiplying a by itself d times is equivalent (mod c) to multiplying b by itself d times - that is, $a \times a \times ... \times a \equiv b \times b \times ... \times b \pmod{c}$, or $a^d \equiv b^d \pmod{c}$. And this is why we are able to "replace" $(a^{p-1})^k$ with 1^k when working (mod p): because $a^{p-1} \equiv 1 \pmod{p}$.]

So we see that if we have a multiple of $p-1$ as a power of a, then we can easily break it down by exponent rules to get that $a^{k \times (p-1)} = a^{(p-1) \times k} = (a^{p-1})^k \equiv 1^k = 1 \mod p$.

Finally, what if our exponent (greater than $p-1$) isn't a perfect multiple of $p-1$? What then? We can no longer break it down as an integer times $p-1$, *but* we can still break it down as an integer times $p-1$ *plus* another integer. For instance, if $p-1=5$ but our exponent is 7, we can write 7 as $7 = 1 * 5 + 2$; if our exponent is 33, we can write $33 = 6 * 5 + 3$. Knowing this, we use exponent rules once again. Exponent rules tell us that

$$a^{b+c} = a^b \times a^c \tag{3}$$

Thus, if we have an exponent $k > p-1$ that can be written as $k = m \times (p-1) + n$, then we can break down a^k using (3) as follows:

$$a^k = a^{m \times (p-1) + n} = a^{m \times (p-1)} \times a^n = a^{(p-1) \times m} \times a^n = (a^{p-1})^m \times a^n$$

and now, using Fermat's Theorem and the rules of modular arithmetic, we have

$$(a^{p-1})^m \times a^n \equiv 1^m \times a^n = a^n \pmod{p}.$$

And hence the motivation for using Fermat's Theorem to find remainders of large powers. Now we will solve the problem at hand. We want to find $12^{11764} \pmod{1471}$ using Fermat's Theorem: $a^{p-1} \equiv 1 \pmod{p}$ for prime p and integers a coprime to p.[1]

First, we need to determine whether we even *can* use Fermat's Theorem. Every theorem has its assumptions, and Fermat's is no different. Let's verify that it can be applied first. Take the number $p = 1471$, which we can confirm is prime. Let $a = 12$. Since $\gcd(12, 1471) = 1$, we can, indeed, apply Fermat's Theorem:

$$12^{1471-1} \equiv 1 \pmod{1471}, \text{ or } 12^{1470} \equiv 1 \pmod{1471}.$$

We then write our original exponent of interest, 11764, as as integer times 1471 plus another integer: $11764 = 8 \times 1470 + 4$.

Finally, using exponent rules we have

$$12^{11764} = 12^{8\times 1470+4} = 12^{8\times 1470}\times 12^4 = 12^{1470\times 8}\times 12^4 =$$
$$\left(12^{1470}\right)^8 \times 12^4 \equiv 1^8 \times 12^4 = 20736 \equiv 142 \pmod{1471}.$$

Thus, the remainder when 12^{11764} is divided by 1471 is 142.

[1] Notice: since p is prime, almost every integer is coprime to p: the only integers that are *not* coprime to p are multiples of p. This gives you an easy way to check if an integer a is coprime to p for using Fermat's Theorem: if it's not divisible by p, then it's coprime to p and Fermat's Theorem applies. Wala!

4) Using Euler's Theorem, determine $99^{29403} \bmod 34643$.

Solution

Euler's Theorem states that for any positive integer n and an integer a that is coprime to n, we have $a^{\varphi(n)} \equiv 1 \pmod{n}$. We apply the same reasoning here that we did with Fermat's Theorem.

Take $n = 34643 > 0$ and $a = 99$. We first determine whether a and n are coprime by using their prime factorizations: $34643 = 7^3 \times 101$ and $99 = 3^2 \times 11$. Since a and n do not share any prime factors, they are coprime and we can apply Euler's Theorem:

$$99^{\varphi(34643)} \equiv 1 \pmod{34643}.$$

Now, we calculate $\varphi(34643)$. Recall that one formula for Euler's totient function is $\varphi(n) = n\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)\ldots\left(1-\frac{1}{p_k}\right)$ for each of the k primes in n's prime factorization. Since $34643 = 7^3 \times 101$, we can use this formula to get

$$\varphi(34643) = 34643 \times \left(1-\tfrac{1}{7}\right)\times\left(1-\tfrac{1}{101}\right) = 7^3\times 101\times\tfrac{6}{7}\times\tfrac{100}{101} = 7^2\times 6\times 100 = 29400.$$[2]

Hence,

$$99^{29400} \equiv 1 \pmod{34643}.$$

We then write 29403 as an integer times 29400 plus another integer: $29403 = 1\times 29400 + 3$.

Finally, using exponent rules we have

$$99^{29403} = 99^{1\times 29400+3} = 99^{29400}\times 99^3 \equiv 1\times 970299 \equiv 295 \pmod{34643}.$$

Thus, $99^{29403} \equiv 295 \pmod{34643}$.

[2] Alternatively, we could have used the multiplicativity of $\varphi(n)$ and the fact that for a prime p and positive integer k, $\varphi(p) = p-1$ and $\varphi(p^k) = p^k - p^{k-1}$:

$$\varphi(34643) = \varphi\left(7^3\times 101\right) = \varphi\left(7^3\right)\times\varphi(101) = \left(7^3-7^2\right)\times 100 = 294\times 100 = 29400.$$

5) Trace through the Miller-Rabin algorithm testing n = 169 for primality using the test value of a = 2. In particular, first state the value $X = 2^d \pmod{169}$, where d is odd that gets calculated in the algorithm. From that point, the algorithm continually squares X, and depending on the new result, either returns "Probably Prime" or "Composite". Show each value of X in the algorithm and when the decision is made what to return and what is returned.

Solution

First, let's take a look at the Miller-Rabin algorithm, based on this semester's scanned notes (if you referred to the typed lecture notes or the written notes from Fall 2019, both under the "CIS 3362 Lecture Notes" page, see the alternate solution below):

```
isPrime(int n)
{
      rewrite  n – 1 = 2^k × n′,  for some odd integer  n′
      pick a random integer  a ∈ [2, n – 1]
      calculate  X = a^n′ (mod n)
      if  X ≡ 1 (mod n)
            return "is probably prime"
      for (i = 0; i < k; i++)
      {
            if  X ≡ n – 1 (mod n)
                  return "is probably prime"
            X = X^2 (mod n)
      }
      return "composite"
}
```

We will trace through this algorithm, with $n = 169$ and $a = 2$.

First, we write $n - 1 = 2^k \times n'$: $169 - 1 = 2^3 \times 21$, or $168 = 2^3 \times 21$. Thus, $k = 3$ and $n' = 21$.

Then, picking $a = 2$, we must calculate $X = a^{n'}$, or $X = 2^{21} \pmod{169}$. Now, 2^{21} is small enough that you can calculate it directly and just use a calculator to find that value (mod 169), which works beautifully. We have $X = 2^{21} = 2097152 \equiv 31 \pmod{169}$, or $X = 31 \pmod{169}$.[3]

Since $X \equiv 31 \neq 1 \pmod{169}$, we do not return "is probably prime" and move into the `for` loop. We will keep track of each value of `i`, which in this case goes from 0 to $k - 1 = 2$.

`i = 0`: $X \equiv 31 \neq 168 \pmod{169}$, $X = X^2 = 31^2 = 961 \equiv 116 \pmod{169}$, `continue`
`i = 1`: $X \equiv 116 \neq 168 \pmod{169}$, $X = X^2 = 116^2 = 13456 \equiv 105 \pmod{169}$, `continue`
`i = 2`: $X \equiv 105 \neq 168 \pmod{169}$, $X = X^2 = 105^2 = 11025 \equiv 40 \pmod{169}$, `continue`

At this point, we exit the `for` loop and return "composite".

Alternate Solution

Here is the Miller-Rabin algorithm as written in Fall 2019 and in the typed lecture notes (the two psuedocodes are equivalent):

```
isPrime(int n)
{
      rewrite n − 1 = 2^k × n′, for some odd integer n′
      pick a random integer a ∈ [2, n − 1]
      if gcd(a, n) ≠ 1
            return "composite"
      calculate X = a^n′ (mod n)
      if X ≡ 1 (mod n)
            return "is probably prime"
      for(i = 0; i < k; i++)
      {
            X = X^2 (mod n)
            if X ≡ −1 (mod n)
                  return "is probably prime"
            if X ≡ 1 (mod n)
                  return "composite"
      }
      return "composite"
}
```

We will trace through this algorithm, with $n = 169$ and $a = 2$.

First, we write $n - 1 = 2^k \times n'$: $169 - 1 = 2^3 \times 21$, or $168 = 2^3 \times 21$. Thus, $k = 3$ and $n' = 21$.

Then, picking $a = 2$, we quickly see that $\gcd(2, 169) = 1$, since 2 is even and 169 is odd. Thus, we do not return "composite".

Next, we must calculate $X = a^{n'}$, or $X = 2^{21} \pmod{169}$. Now, 2^{21} is small enough that you can calculate it directly and just use a calculator to find that value $\pmod{169}$, which works beautifully. We have $X = 2^{21} = 2097152 \equiv 31 \pmod{169}$, or $X = 31 \pmod{169}$.[3]

Since $X \equiv 31 \neq 1 \pmod{169}$, we do not return "is probably prime" and move into the `for` loop. We will keep track of each value of `i`, which in this case goes from 0 to $k - 1 = 2$.

`i = 0`: $X = X^2 = 31^2 = 961 \equiv 116 \pmod{169}$, $X \neq -1$ or $1 \pmod{169}$, `continue`
`i = 1`: $X = X^2 = 116^2 = 13456 \equiv 105 \pmod{169}$, $X \neq -1$ or $1 \pmod{169}$, `continue`
`i = 2`: $X = X^2 = 105^2 = 11025 \equiv 40 \pmod{169}$, $X \neq -1$ or $1 \pmod{169}$, `continue`

At this point, we exit the `for` loop and return "composite".

[3] Just to show you another way to do it (this one's called fast mod exponentiation, which you'll learn in the next section), I noticed that $2^7 = 128$, which is closer to 169 than $2^8 = 256$. We remap $2^7 = 128$ to a smaller absolute value (mod 169): $2^7 = 128 \equiv -41 \pmod{169}$. Then, $2^{21} = \left(2^7\right)^3 \equiv (-41)^3 = -68921 \equiv 31 \pmod{169}$. We see that the purpose of doing it this way is that $(-41)^3$ is a smaller calculation than 2^{21}, hence the term fast mod exponentiation.

6) Use Fermat Factoring to factor 37001. Fill in the table below (it's possible that more rows than necessary are included.) and then provide the factorization.

X	X^2 - 37001	Perfect Square?

Solution

The amazing idea behind Fermat factoring is the difference of squares: $n = x^2 - y^2 = (x - y)(x + y)$. Given an odd composite integer n, if we can write it as a difference of two squares, x^2 and y^2, then we can factor it into $n = (x - y)(x + y)$.

Here, we have $n = 37001$. Since the question is asking us to use Fermat factoring to factor it, we can infer that the conditions of Fermat factoring are being met - namely, that 37001 is an odd composite integer.

First, we rewrite $n = x^2 - y^2$ as $x^2 - n = y^2$. This shows us that we need to find an integer x such that $x^2 - n$ gives us the perfect square of another integer. What we will do is take increasing values of x, calculate $x^2 - n$, and then calculate $\sqrt{x^2 - n}$. If this square root gives us an integer, we can Fermat factor n. Otherwise, if it gives us a decimal, we'll continue with the next value of x.

We begin with the smallest integer x whose square, x^2, is greater than $n = 37001$. Since $\sqrt{37001} \approx 192.3$, we start with $x = 193$. Indeed, $x^2 = 193^2 = 37249$, which is the smallest square greater than 37001. We calculate $193^2 - 37001 = 248$, and $\sqrt{248} \approx 15.74$, which is not an integer. Thus, $193^2 - 37001$ is not a perfect square and we move on.

X	X^2 - 37001	Perfect Square?
193	248	No

Next, we have $x = 194$. We calculate $194^2 - 37001 = 635$, and $\sqrt{635} \approx 25.20$, which is not an integer. Thus, $194^2 - 37001$ is not a perfect square and we move on.

X	X^2 - 37001	Perfect Square?
193	248	No
194	635	No

Next, we have $x = 195$. We calculate $195^2 - 37001 = 1024$, and $\sqrt{1024} = 32$, which *is* an integer! Wala! $194^2 - 37001$ is a perfect square, and we can now Fermat factor 37001.

X	X^2 - 37001	Perfect Square?
193	248	No
194	635	No
195	1024	Yes!

To recap, we have $195^2 - 37001 = 1024 = 32^2$, or $195^2 - 37001 = 32^2$. Rearranging, we have $37001 = 195^2 - 32^2$. Using our difference of squares formula, we have

$$37001 = 195^2 - 32^2 = (195 - 32)(195 + 32) = 163 \times 227,$$

or $37001 = 163 \times 227$. Finally, checking with a calculator, we can verify this is true.

7) Define f(n) = min($\frac{\varphi(k)}{k}$), for all positive integers $2 \leq k \leq n$. For example, f(32) = $\frac{4}{15}$, because for each integer in between 2 and 32, the smallest possible value of $\frac{\varphi(k)}{k}$ is $\frac{\varphi(30)}{30} = \frac{8}{30} = \frac{4}{15}$. Write a program that asks the user for the input value of n, and then prints out f(n) as a fraction in lowest terms. You may assume that the user won't enter a number bigger than 3×10^8. (Note: You can solve this problem in different ways. A brute force solution will be accepted, but there is a much more elegant solution that runs really fast.) Please write your program in C, Java or Python and attach it separately.

Solution - Brute Force

The brute force solution goes as follows: given an integer n less than 3×10^8, we iterate through all integers from 2 to n and find $\frac{\varphi(k)}{k}$. We keep track of the minimum such value of $\frac{\varphi(k)}{k}$ as we iterate, and in the end, we print the absolute minimum in lowest terms.

Of course, this is looking at it from a high level; on a low level, there are many things we need to calculate and keep track of. For instance, finding $\varphi(k)$ requires finding the prime factors of k and then using one of the formulas for $\varphi(k)$ to calculate it. Furthermore, the question requires that the output be in lowest terms, so there also needs to be a way to reduce $\frac{\varphi(k)}{k}$.

Pseudocode would look something like this:

```
findPrimes(x)
{
     // finds the prime factors of x
}

reduceFraction(y)
{
     // reduces the fraction y to lowest terms
}

min = φ(2)/2

for (i = 2 to n)
{
     // use findPrimes() to calculate φ(k)
     val = φ(k)/k
     if (val < min)
          min = val
}

result = reduceFraction(val)
print(result)
```

Solution - Elegant

Arup is a sneaky final boss, isn't he? It wasn't going to be easy coming this far.

Yes, all the rumors are true: there is an elegant solution that finds $f(n)$ lightning quick. But to get there, we must embark on a journey more dangerous than before. We must first make a few stops on this journey to defeat the final boss Arup.

Let k be the integer between 2 and n that minimizes $\frac{\varphi(k)}{k}$. We begin by prime factoring k. Let's say k has the prime factorization

$$k = {p_1}^{a_1} \times {p_2}^{a_2} \times ... \times {p_m}^{a_m},$$

where p_1, p_2, …, p_m are the m distinct prime factors of k and a_1, a_2, …, a_m are the corresponding exponents of those prime factors.

Now, the key to this clever chaos lies in $\varphi(k)$. Recall the formula

$$\varphi(k) = k \times \left(1 - \frac{1}{p_1}\right) \times \left(1 - \frac{1}{p_2}\right) \times ... \times \left(1 - \frac{1}{p_m}\right),$$

where p_1, p_2, …, p_m are the m distinct prime factors of k.

Using this formula, we can rewrite $\frac{\varphi(k)}{k}$ as

$$\frac{\varphi(k)}{k} = \frac{k \times \left(1-\frac{1}{p_1}\right) \times \left(1-\frac{1}{p_2}\right) \times ... \times \left(1-\frac{1}{p_m}\right)}{k} = \left(1 - \frac{1}{p_1}\right) \times \left(1 - \frac{1}{p_2}\right) \times ... \times \left(1 - \frac{1}{p_m}\right),$$

where p_1, p_2, …, p_m are the m primes in the prime factorization of k. Remember this new product version of $\frac{\varphi(k)}{k}$ - we'll refer to it throughout. Then we have

$$f(n) = \min\left(\frac{\varphi(k)}{k}\right) = \min\left(1 - \frac{1}{p_1}\right) \times \left(1 - \frac{1}{p_2}\right) \times ... \times \left(1 - \frac{1}{p_m}\right).$$

In other words, we want to minimize the product $\left(1 - \frac{1}{p_1}\right) \times \left(1 - \frac{1}{p_2}\right) \times ... \times \left(1 - \frac{1}{p_m}\right)$. We need to find the value of k that will minimize this product for any n. How will we do that? We will begin by making three stops - three important observations on minimizing this product. After that, we'll put those observations together, ultimately leading us to find $f(n)$ efficiently for any n.

If you choose to come with me on this journey, I promise we will discover many exciting things.

So you're in? Sweet! Oh no, no, you won't need all that - just a willingness to learn and the perseverance to walk through strong winds. We'll make it together, my friend. Now off we go!

Here we are at our first stop. First, look back at our rewritten version of $\frac{\varphi(k)}{k}$. You'll notice that the product is only dependent on the prime factors of k - it is *not* dependent on the exponents of

those primes. Thus, to minimize $\frac{\varphi(k)}{k}$, we set all of the exponents in k's prime factorization to the lowest possible value: 1. (We cannot set them to 0 - can you see why?) That is, we set

$$a_1 = a_2 = ... = a_m = 1 .$$

Great job! That wasn't too bad. Get ready, now - we have a monster coming up at our next stop.

Up ahead is our second observation. Notice that each prime p_1, p_2, …, p_m is positive, and 1 divided by each of these primes is also positive: $\frac{1}{p_1}$ is still positive, $\frac{1}{p_2}$ is still positive, and so on. And since each $\frac{1}{p_i}$ is less than 1 and also being subtracted from 1, that means that each of the terms in the product above — each $1 - \frac{1}{p_i}$ term — is greater than 0 and less than 1.

Before I go into why this is important, let's first take a detour: what does it mean - intuitively, conceptually - to multiply a number by a fraction between 0 and 1? When we multiply 20 by $\frac{3}{4}$, we are finding "three-fourths of twenty", or a fraction of 20. Similarly, to divide a pie between 6 people, we multiply its area by $\frac{1}{6}$ to find "one-sixth of the pie", or a fraction of the whole pie. Essentially, we are just taking a fraction of the whole - a smaller part - so every time we multiply a number by a fraction between 0 and 1, the original number gets smaller.

Now what happens when we multiply a fraction between 0 and 1 by another fraction between 0 and 1? Same thing, really: we still get a smaller number. Multiplying $\frac{2}{3}$ by $\frac{7}{8}$ means we are taking "two-thirds of seven-eighths" - not "one whole of seven-eighths", but "two-thirds" of it. We are taking a fraction of $\frac{7}{8}$, and because of that it results in a number less than $\frac{7}{8}$. The original number still gets smaller.

Alright, now let's circle back to the main part of our stop. Remember our product $\left(1 - \frac{1}{p_1}\right) \times \left(1 - \frac{1}{p_2}\right) \times ... \times \left(1 - \frac{1}{p_m}\right)$? From our analysis three paragraphs ago, we know that this product is completely made of fractions that are between 0 and 1. That means when we multiply $1 - \frac{1}{p_1}$ by $1 - \frac{1}{p_2}$, we get a smaller number. When we multiply that result by $1 - \frac{1}{p_3}$, we get an even smaller number. Multiply *that* by yet another fraction, say $1 - \frac{1}{p_4}$, and we get an even *smaller* number. And so on and so forth.

And now remember that we want to minimize this product. Our analysis above show us that the more $1 - \frac{1}{p_i}$ terms we multiply, the smaller the product $\left(1 - \frac{1}{p_1}\right) \times \left(1 - \frac{1}{p_2}\right) \times ... \times \left(1 - \frac{1}{p_m}\right)$ gets. Thus, to minimize $\frac{\varphi(k)}{k}$, we want it to have as many $1 - \frac{1}{p_i}$ terms as possible. And that is our second observation! We have overcome the monster that it was. **DAMAGE 5000!!!**

Let's take a break, catch our breaths, drink some water. Don't worry if this all seems jumbled right now - in the end, it will all come together. You're doing great! I really appreciate you keeping up and trying your best. And now we're at our last stop, our last observation.

So far, we've found that to minimize $\frac{\varphi(k)}{k}$, we need k to have all of the exponents of its prime factorization equal to 1 and also have as many prime factors as possible. But how about the prime factors themselves? Which prime factors does k need to have? How do we pick?

First, remember that $\frac{\varphi(k)}{k} = \left(1 - \frac{1}{p_1}\right) \times \left(1 - \frac{1}{p_2}\right) \times ... \times \left(1 - \frac{1}{p_m}\right)$, so if we want to minimize the product, we have to minimize each of the terms inside of that product. That is, we want the smallest $1 - \frac{1}{p_i}$ terms possible.

But that doesn't directly tell us about the prime factors. We need to find a way to get from the terms $1 - \frac{1}{p_i}$ to the prime factors p_i of k.

Here's what we'll do. Let's take two positive primes, p and q. Let's assume that $1 - \frac{1}{p} < 1 - \frac{1}{q}$. In other words, $1 - \frac{1}{p}$ is the smallest term between the two terms. From that, we will derive a statement comparing p and q. There are two possible cases:

> Case 1: If we derive that $p < q$, then our implication is that $1 - \frac{1}{p} < 1 - \frac{1}{q} \quad \Rightarrow \quad p < q$. This will tell us that to get the smallest $1 - \frac{1}{p_i}$ terms, we will need the <u>smallest</u> prime factors p_i.
>
> Case 2: If we derive that $p > q$, then our implication is that $1 - \frac{1}{p} < 1 - \frac{1}{q} \quad \Rightarrow \quad p > q$. This will tell us that to get the smallest $1 - \frac{1}{p_i}$ terms, we will need the <u>largest</u> prime factors p_i.

Let's start our derivation. We begin with our assumption: $1 - \frac{1}{p} < 1 - \frac{1}{q}$. We first subtract 1 from each side, giving us $-\frac{1}{p} < -\frac{1}{q}$. Then, we multiply by -1 on both sides, which flips the inequality and gives us $\frac{1}{p} > \frac{1}{q}$. Finally, we multiply by pq on both sides (which is positive since p and q are positive, so the inequality will not need to be flipped), giving us $\frac{pq}{p} > \frac{pq}{q}$, or $q > p$, or $p < q$.

We have arrived at Case 1. Thus, to get the smallest $1 - \frac{1}{p_i}$ terms and thus minimize $\frac{\varphi(k)}{k}$, we will need the smallest prime factors of k. And bim, boom, pow! Behold, our final observation.

Wow! We made it through our three stops and we're getting real close now. Take a breath, yes. Let's rest here for a night - we deserve it. When we put it all together... I think we'll have enough to take down our final boss. I believe it. We're coming for you, Arup!

Let's review our three important observations. To minimize $\frac{\varphi(k)}{k}$, we need k such that

1. the exponents in its prime factorization are all equal to 1,
2. it has as many prime factors as possible, and
3. it has the smallest prime factors in its prime factorization.

Thus, our k will be $k = 2^1 \times 3^1 \times 5^1 \times ...$ — wait, where do we stop? Just how many is "as many prime factors as possible"? Oh no, we're missing something… Do we have to take another stop?

Well, if you go back to the definition of $f(n)$, you'll see that k is a number between 2 and n, so it cannot exceed n. Thus, we stop multiplying prime factors once k is as large as possible without exceeding n. For instance, if $n = 1000$, then k will have all the smallest prime factors up to 7, since $2 \times 3 \times 5 \times 7 = 210$ and multiplying that by the next prime, 11, gives us a number bigger than $n = 1000$.

That was simple, wasn't it? No worries, we didn't have to take another stop. Did I trick you? Haha I'm just messing with ya.

Alright, let's get back to it! I know, I know - I was the one who started all the shenanigans. Anyway... We still have to take down our final boss. We're almost there!

We now know that k will be the product of successive prime numbers starting from 2 and going all the way up until $k > n$. Now all we need to do is calculate $\frac{\varphi(k)}{k}$ and reduce it to lowest terms. We need to have a way of reducing *any* fraction to lowest terms. How can we do that?

How about let's reduce a fraction together, say $\frac{24}{84}$. One way to do this would be trial division: notice that the 24 and 84 are both even, so we can divide them by 2 to get $\frac{12}{42}$, which we can divide by 2 again by the same logic to get $\frac{6}{21}$. At this point, we can see that 6 and 21 both share a factor of 3, so we divide them both by 3 to get $\frac{2}{7}$, which is the final reduced form.

Alternatively, we could have immediately noticed that 24 and 84 share a common factor of 12, and $\frac{24}{84} \div \frac{12}{12} = \frac{2}{7}$, which shows us that 12 is the greatest common factor of 24 and 84.

Well, well, well... if it isn't Greatest Common Divisor. Do you remember ole GCD? Yes yes, we have to get past them, too. Indeed, it is the greatest common divisor that we will use to reduce $\frac{\varphi(k)}{k}$ to lowest terms. But no worries - we've seen them before, and we have the right tools to handle them again. Euclidean algorithm, I summon you!

Finally, let's gather all of the tools we need to create our program. We need the input n, the minimizer k, a list to keep track of the prime factors of k, and a list of all of the prime numbers whose product altogether is the greatest possible product that does not exceed 3×10^8 (plus one more to be safe). We'll also need a way to calculate $\varphi(k)$ using its prime factors, as well as a way to calculate the gcd() of $\varphi(k)$ and k.

And alas, the pseudocode for this solution, below. You'll find my full implementation for this solution in a separate `.py` file that comes with this solution in the Homework Assignments page. It's heavily commented and documented to make sure you can easily follow along.

```
n = input
k = 1
kPrimeFactors = []
primes = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29]

phi(primeFactors)
{
     // given a list of prime factors, primeFactors,
     // whose exponents are all 1 in the prime
     // factorization of a number, k, this will return φ(k)
}

gcd(a, b)
{
     // uses the euclidean algorithm to return gcd(a, b)
}

for (each prime in primes)
{
     if (k * prime > n)
     {
          // use phi() to calculate φ(k)
          // use gcd() to reduce φ(k)/k to lowest terms
          print (φ(k)/k in lowest terms)
          break
     }

     k = k * prime
     // add prime to kPrimeFactors
}
```

Looking at the pseudocode, you'll realize just how elegant this solution is: by making a few important observations, we were able to cut down from brute forcing through potentially 300 million numbers (not to mention finding *each* of their primes) to iterating through just ten! With this solution, we only need to iterate through the list of primes whose product does not exceed 3×10^8, which was a list of 10 primes.

And finally! We have the sword-ocode to finally defeat the almighty final boss Arup! *Draws sword-ocode from its sheath* Down with it! We fight! *Mighty sword sounds*

DAMAGE 1000000!!!!! DEFEAT!!!

We've defeated Arup! AAA I'm so proud of you! You conquered many stops and monsters on this journey, and you battled with everything in you. I am so honored to know you, and I hope you learned as much as I did on our journey together. Thank you for coming along. This was fun!

Let's do this again next homework?