Math 300 B
Spring 2017

Some remarks on the Euler totient function

These are a few scattered remarks about the Euler totient function, discussed in class on Wednesday, May 03. First, recall the main concept being discussed here:

Definition 1. The *Euler totient function* $\phi : \mathbb{N} - \{1\} \to \mathbb{N}$ is the function defined by setting $\phi(n)$ to be the number of elements in $\{0, \cdots, n-1\}$ that are coprime to n. ♦

Now suppose your natural number $n \geq 2$ decomposes into prime factors as

$$n = \prod_{i=1}^{k} p_1^{t_1} \cdots p_k^{t_k} \tag{1}$$

for primes $p_1 < \cdots < p_k$ and positive integers t_i. Then, as an application of the Inclusion-Exclusion Principle, the textbook proves

Theorem 2. *For n as in (1) the Euler totient $\phi(n)$ is*

$$\sum_{j=0}^{k} (-1)^j \sum_{p_{i_1} < \cdots < p_{i_j}} \frac{n}{p_{i_1} \cdots p_{i_j}}. \tag{2}$$

■

That looks a bit complicated, but all it's saying is that you take n, then subtract from it all quotients $\frac{n}{p_i}$, then add to that all quotients $\frac{n}{p_i p_j}$, then subtract from *that* all $\frac{n}{p_i p_j p_l}$, etc. Here's how to say all of this in more compact form (this is also problem 10.30 in your book).

Corollary 3. *For n as in Theorem 2 we have*

$$\phi(n) = n \cdot \prod_{i=1}^{k} \left(1 - \frac{1}{p_i}\right). \tag{3}$$

Proof. Exercise; just use simple arithmetic to translate (2) into (3). ■

Now, here's an interesting way to convince yourself that (3) is the right answer, albeit using some material we haven't covered: you can use a probabilistic argument.

First, think of $\frac{\phi(n)}{n}$ as the probability that, when you choose a number among $\mathbb{Z}_n = \{0, \cdots, n-1\}$ at random, you hit upon one that is coprime to n. Now, how else could you compute that probability?

Well, it's the probability that your random number is

(1) not divisible by p_1;
(2) not divisible by p_2;
(3) not divisible by p_3;
(4) etc.

Now, the probability that (1) happens is 1 minus the probability that your number *is* divisible by p_1. Since there are $\frac{n}{p_1}$ numbers in $\mathbb{Z}_n$ that are divisible by p_1, this probability is $1 - \frac{1}{p_1}$.

The same reasoning goes for the other primes, giving you $1 - \frac{1}{p_i}$ as the probability that your randomly-chosen number from $\mathbb{Z}_n$ is *not* divisible by p_i. All in all, because being divisible by p_i is independent from being divisible by p_j for $i \neq j$, the probability that *all* events of the form (not divisible by p_i) happen simultaneously is the product

$$\prod_{i=1}^{k} \left(1 - \frac{1}{p_i}\right).$$

Remark 4. The above is not meant as a rigorous proof, because we haven't covered the formalism of probability theory. It is only meant as a heuristic that (hopefully) appeals to your intuition. ⧫

Here's another cool bit of info; this is just me dropping a bunch of buzzwords in case you're curious and want to look up any of this, as well as an attempt to provide some justification for my claim that this sort of mathematical content has wide applicability and deep practical implications.

If you read about the RSA cryptography algorithm (one of the most popular cryptographic techniques), you'll see that it essentially consists of the following:

- pick a number $n = pq$ that is the product of two distinct primes;
- pick a number $0 < e < \phi(n) = (p-1)(q-1)$ that is coprime to $\phi(n)$;
- the *public key* that you make available to anyone wanting to send you a secret message is the pair of integers n and e;
- the *private key* that you keep to yourself to decode the secrets you receive is the *multiplicative inverse* d of e modulo $\phi(n)$: a number $0 < d < \phi(n)$ such that $\phi(n) | de-1$.

I am oversimplifying massively; look up RSA cryptography for more info (which is widely available; the Wikipedia page is a good start). I am only trying to make the point that the topics we have been looking into recently are very close, mathematically, to some of the fundamental machinery powering internet communication today.

The reason why the RSA algorithm is effective is that even though you give away your $n = pq$, it is difficult for an attacker to pick it apart and factor it (i.e. find p and q, which are gigantic primes in applications), and hence to find $\phi(n) = (p-1)(q-1)$ and compute the inverse of e modulo that number.

So this ties in with the point I was making in class on Wednesday, May 03: you can't express $\phi(n)$ simply in terms of n alone, you need to know the prime decomposition of n, and then express $]phi(n)$ in terms of those primes instead.