# Homework 7

Due 06/21/17

June 15, 2017

Euler's Totient function (or Euler's Phi function) is a function that accepts a positive integer $n$ and returns how many positive integers less than $n$ are *relatively prime* to $n$. (That is, they have greatest common divisor of 1 with $n$.)

As an example, consider computing $\phi(12)$. We can see from the following table that only 1, 5, 7, and 11 are relatively prime to 12, so $\phi(12) = 4$.

| $x$ | $\gcd(x, 12)$ |
|---:|---:|
| **1** | **1** |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| **5** | **1** |
| 6 | 6 |
| **7** | **1** |
| 8 | 4 |
| 9 | 3 |
| 10 | 2 |
| **11** | **1** |

The naïve EulerPhi algorithm below computes Euler's Phi recursively based on the following recurrence:

$$\phi(n) = \begin{cases} n-1, & \text{if } n \text{ is prime or } 1 \\ \frac{\phi(a)\phi(b)\gcd(a,b)}{\phi(\gcd(a,b))}, & \text{where } a \text{ is a factor of } n \text{ and } b = n/a \end{cases}$$

**Input**: $n$: positive integer
**Output**: $\phi(n)$

```
Algorithm: EulerPhi
for a = 2 to ⌊√n⌋ do
    if a divides n then
        b = n/a
        g = gcd(a, b)
        return EulerPhi(a) · EulerPhi(b) · g/EulerPhi(g)
    end
end
return n − 1
```

1. Describe a small example for why this algorithm is a good candidate for dynamic programming. *Hint*: pick a number with multiple factors. You may use Wolfram Alpha (or some other calculator) to compute gcd's.

2. What data structure would you use in order to store the solutions to subproblems?

3. What is a reasonable sentinel value to indicate that the algorithm has not yet computed $\phi(n)$?

4. Give pseudocode for a memoized implementation of this algorithm. You may assume that someone has implemented a function $\gcd(m, n)$ that computes the gcd of two integers.