WORKSHEET 7

The Euclidean Algorithm

Given a collection $a_1, a_2, \ldots, a_n$ of positive integers, their *greatest common divisor*, or *gcd*, is the largest positive number d dividing all of them. For instance, $\gcd(105, 140) = 35$ because $105 = 35 \times 3$ and $140 = 35 \times 4$, and no number larger than 35 divides both 105 and 140.

DEFINITION. If $\gcd(a, b) = 1$, then we say that a and b are *relatively prime*.

PROBLEM 7.1. Suppose we know the prime factorizations of a and b: $a = 2^{e_2}3^{e_3}5^{e_5}\cdots$ and $b = 2^{f_2}3^{f_3}5^{f_5}\cdots$. Express $\gcd(a, b)$ in terms of the e_i's and f_i's.

Closely related the gcd is the *least common multiple*, or *lcm*. The lcm of $a_1, \ldots, a_n$ is the smallest positive integer divisible by each of the a_i's.

PROBLEM 7.2. Show that if a and b are positive integers, then $ab = \gcd(a, b)\,\mathrm{lcm}(a, b)$.

Based on Problem 7.1, you know how to compute the gcd of a and b provided that you are given the prime factorizations of a and b. But if a and b are large, then computing the prime factorizations of a and b might be a difficult problem. So, it would be nice if there were a way of computing the gcd without knowing the prime factorizations. This seems unlikely at first, because $\gcd(a, b)$ seems to encode some facts about the prime factorizations, but remarkably, this is possible, and this is the content of the Euclidean algorithm.

PROBLEM 7.3. Show that if a and b are positive integers with $b > a$, then $\gcd(a, b) = \gcd(a, b - a)$.

PROBLEM 7.4. More generally, suppose that $b > ak$. Show that $\gcd(a, b) = \gcd(a, b - ak)$.

PROBLEM 7.5. Use the result of the previous problem to compute $\gcd(590413, 830581)$ by hand, in a reasonable amount of time.

What you presumably did in the previous problem was to run the following algorithm, known as the *Euclidean algorithm*. The notation $\lfloor x \rfloor$ means the greatest integer $\leq x$, called the *floor* of x.

Euclidean algorithm

```
function gcd(a,b)
    if a > b then
        swap a and b
    if a = 0 then
        return b
    k ← ⌊b/a⌋
    return gcd(a, b − ak)
```

PROBLEM 7.6. Use the Euclidean algorithm to compute $\gcd(233, 377)$. Why does it take so long?

PROBLEM 7.7. Show that if x and y are integers, then $ax + by$ is a multiple of $\gcd(a, b)$.

We would now like to show the reverse: that any multiple of $\gcd(a, b)$ can be written as $ax + by$ for some (not necessarily positive) integers x and y.

PROBLEM 7.8. Show that it suffices to check that $\gcd(a, b)$ can be expressed in the form $ax + by$, where x and y are integers.

In order to show that $\gcd(a, b)$ can be written as $ax + by$, we need to take a more careful look at the Euclidean algorithm. Let us assume that $a \leq b$. Let $b_0 = b$ and $b_1 = a$. If, for some n, we have $b_n = 0$, then $\gcd(a, b) = b_{n-1}$. If $b_n \neq 0$, then let k be the unique positive integer such that $0 \leq b_{n-1} - kb_n < b_n$. Let $b_{n+1} = b_{n-1} - kb_n$.

PROBLEM 7.9. For $a = 35$ and $b = 91$, compute all the b_n's.

PROBLEM 7.10. Show that each b_n can be written as $b_n = ax + by$ for some integers x and y.

PROBLEM 7.11. Conclude that there exist integers x and y such that $\gcd(a, b) = ax + by$.

This result is a very important fact in number theory, called *Bézout's Lemma*. An interesting application is a new categorization of prime numbers.

PROBLEM 7.12. Show that if p is prime, and a and b are integers such that ab is a multiple of p, then at least one of a and b is already a multiple of p.

PROBLEM 7.13. Show that if n is composite, then there exist integers a and b such that neither a nor b is a multiple of n, but ab is a multiple of n.

The Euclidean Algorithm

Alexander Friesen 11/4-9/25

7.1: For each prime factor shared, the one with the lower power is chosen because it factorizes the greater and both numbers - which is criteria for the GCD. Express this as $\min(2^{e_2}, 2^{f_2})$ or $(3^{e_3}, 3^{f_3})$ etc. The GCF of (a, b) is $2^{(\min(2^{e_2}, 2^{f_2})} \cdot 3^{\min(3^{e_3}, 3^{f_3})} \cdot 5 \cdots$ until the greatest common factor is reached.

7.2: Because the GCF takes the minimum exponent of each factor while the LCM takes the maximum, this just ends up taking every factor when the two are multiplied together, just like $a \cdot b$.

7.3: Suppose that the $\gcd(a, b) \neq \gcd(a, b-a)$. But if $\gcd(a, b)$ divides a and b, then it must divide $b-a$. Likewise, $\gcd(a, b-a)$ must also divide b as it divides $a+(b-a)=b$. But this means that $\gcd(a, b) = \gcd(a, b-a)$. Therefore, the two are equal.

7.4: Repeating the above proof k times gives the same logic that $\gcd(a, b) = \gcd(a, b-ka)$

7.5: Because gcd(a,b) = gcd(a, b-a), plugging in 590413 and 830581 can still make finding the gcd relatively simple. The steps: (590413, 830581) → (240168, 590413) → (240168, 350245) → (110077, 240168) → (110077, 130091) → (20014, 110077) → (20014, 90063) → (20014, 70049) → (20014, 50035) → (20014, 30021) → (10007, 20014) → (10007, 10007) → ... and we're done! GCD = 10007!

7.6: Here we go! (233, 377) → (144, 233) → (89, 144) → (55, 89) → (34, 55) → (21, 34) → (13, 21) → (8, 13) → (5, 8) → (3, 5) → (2, 3) → (1, 2) → (1, 1). This takes so long because 233 and 377 are relatively prime - their GCD = 1, and it takes the longest to compute the Euclidean Algorithm for relatively prime numbers.

7.7: We already know that $a \cdot b$ must be a multiple of gcd(a,b) because it divides both a and b evenly. So if we add any more a's or b's (equivalent to multiplying a or b by x or y), the GCD will still divide $ax \cdot by$ evenly.

7.8: If the GCD can be expressed as $ax+by$, then any multiple of the GCD of a and b can be expressed as $ax+by$ [simply by increasing the x- and y-values.)

7.9: Running the Euclidean Algorithm finds all b_n's:

$(35, 91) \to (35, 56) \to (21, 35) \to (14, 21) \to (7, 14) \to (7, 7)$.

$b_1\ b_0 \quad b_1\ b_2 \quad b_3\ b_1 \quad b_4\ b_3 \quad b_5$

So $b_0 = 91$, $b_1 = 35$, $b_2 = 56$, $b_3 = 21$, $b_4 = 14$, and $b_5 = GCD = 7$. Here, $n = 6$ so that $b_n = 0$, making $b_{n-1} = b_5 = 7$.

7.10: A key aspect of the Euclidean Algorithm is that every b_n (where $n \geq 0$) is divisible by the GCD of a and b - otherwise it wouldn't work. Of course, that also means it is divisible by a certain $ax+by$. Note that x and y are integers: they can be negative, which allows numbers only available by subtraction in the algorithm to count as b_n's that equal $ax+by$. An example is $\gcd(16, 34) = 2 = 16 \cdot -2 + 34 \cdot 1 = 34 - 32 = 2$.

7.11: If every b_n can be $ax+by$, and every $\gcd(a,b)$ can be written as b_{n-1}, then every $\gcd(a,b)$ can be written as $ax+by$.

7.12: If ab is a multiple of p, then ab's prime factorization must include p. But, because p is prime, it must be a factor of a, b, or both. This means that at least one of a and b must be a multiple of p by itself. (This only works if p=prime)

7.13: Say that $n = p^a \cdot p_2^{a_2} \cdot p_3^{a_3} \cdots$ is a prime factorization. This list of $p_i^{a_i}$'s can be divided into two sets a and b, in which:

- a and b are factors of n
- a and b are not multiples of n
- When multiplied together, $ab=n$ unless a-values are changed, in which ab is a multiple of n.

The third property is the answer - there will always be the integers a and b that satisfy the conditions!