CSC 236 Tutorial 9

Harry Sha

July 19, 2023

The first algorithm?

In this tutorial, we'll see the Euclidean Algorithm - one of the oldest algorithms dating back to 300BC.

GCD

The greatest common divisor for two natural numbers m, n is the greatest natural number $a > 1$ that divides both m and n.

a divides n if there is some $k \in \mathbb{N}$ such that $ak = n$.

- What is GCD(6, 15) 3
- What is GCD(5, 10) 5
- What is GCD(0, 19) 19
- What is GCD(4321234,1234321)??? (Don't spend too much time on this) 1

Euclidean Algorithm

The Euclidean Algorithm finds the GCD of two numbers.

Euclidean Algorithm Recursive

```
function gcd(a, b)
    if b = 0
        return a
    else
        return gcd(b, a mod b)
```

Precondition, $a, b \in \mathbb{N}$ with $a \geq b$.

Source: Wikipedia.

Trace

Exercise: Trace this algorithm for GCD(4321234,1234321).

You can search $x \mod y$ on google to get $x \mod y$.

Here's a start:

a	b	a mod b
4321234	1234321	618271
1234321	618271	616050
618271	616050	2221

Solution

a	b	a mod b
4321234	1234321	618271
1234321	618271	616050
618271	616050	2221
616050	2221	833
2221	833	555
833	555	278
555	278	277
278	277	1
277	1	0
1	0	

Correctness

Prove the function on the previous slide is correct.

Lemma

Suppose $a, b \in \mathbb{N}$ with $a \geq b$.

1. GCD(a, b) $=$ GCD(b, a - b)
2. GCD(a, b) $=$ GCD(b, a mod b)

Proof of Lemma part 1

Claim. $\mathrm{GCD}(a, b) = \mathrm{GCD}(b, a - b)$.

Hint. Let $g = \mathrm{GCD}(a, b)$, $g' = \mathrm{GCD}(b, a - b)$. Show that in fact g divides $a - b$ and g' divides a.
Let $g = \mathrm{GCD}(a, b)$. We have $mg = a, ng = b$ since g divides both a and b. Then $a - b = g(m - n)$. Thus, g divides $a - b$.

Let $g' = \mathrm{GCD}(b, a - b)$. then we have $og' = b$, and $pg' = a - b$. Adding the two equations we get $og' + pg' = b$, so $a = g'(o - p)$ and thus g' is a divisor of g'

Since g divides $a - b$ and b, and $g' = \mathrm{GCD}(b, a - b)$, we have $g \leq g'$. Similarly, since g' divides a and b, and $g = \mathrm{GCD}(a, b)$, $g' \leq g$. Thus $g' = g$.

Proof Lemma part 2

Hint. Use induction
P(n). For all $n \in \mathbb{N}$ if $a - nb \in \mathbb{N}$, then
$\mathrm{GCD}(a, b) = \mathrm{GCD}(b, a - nb)$

Sketch. Base case is $\mathrm{GCD}(a, b) = \mathrm{GCD}(b, a)$ which is true.

Inductive step: Use part a of the lemma!

Proof of Correctness

Hint. Use induction on the second argument.

Sketch.

$P(n)$. For all $a \in \mathbb{N}$, the algorithm works on input (a, n). We'll show $\forall n \in \mathbb{N}. P(n)$ by induction.

Base case. This is true b/c $\mathrm{GCD}(a, 0) = a$ for all $a \in \mathbb{N}$.

Inductive step. Use the lemma!

Runtime

Extra: Find the runtime of the algorithm.

Euclidean Algorithm Iterative

Precondition, $a, b \in \mathbb{N}$ with $a \geq b$.

```
function gcd(a, b)
    while b ≠ 0
        t := b
        b := a mod b
        a := t
    return a
```

Source: Wikipedia.

Prove the algorithm is correct

It might help to think of the following questions.

- Postcondition?
- Descending Sequence?
- Loop Invariant?

Proof

Loop invariant. $P(n)$: After the nth iteration.

1. $\mathrm{GCD}(a_n, b_n) = \mathrm{GCD}(a_0, b_0)$.
2. a_n, b_n are natural numbers.

Initialization. $P(0)$ is true b/c both the RHS and the LHS are $\mathrm{GCD}(a_0, b_0)$.

Maintenance. Assume $P(k)$. We'll show $P(k+1)$. The variables b and a are updates as follows: $b_{k+1} = a_k \mod b_k$, and $a_{k+1} = b_k$. Then,

Proof

$$\begin{aligned}
\mathrm{GCD}(a_{k+1}, b_{k+1}) &= \mathrm{GCD}(b_k, a_k \mod b_k) \\
&= \mathrm{GCD}(b_k, a_k) && \text{(Lemma)} \\
&= \mathrm{GCD}(b_0, a_0) && (P(k))
\end{aligned}$$

Termination. We'll show the algorithm terminates later. For now, suppose the while check fails at the start of iteration k. Then $b_k = 0$. By $P(k)$, we have $\mathrm{GCD}(a_k, 0) = \mathrm{GCD}(a_0, b_0) = a_k$, which is the value that we return. Thus, we return $\mathrm{GCD}(a_0, b_0)$ as required.

Descending sequence. We claim that b_i forms a descending sequence. Pick any i. The above proof shows that a_i and b_i are both natural numbers. If $b_i = 0$, the loop terminates. Otherwise,

Proof

$b_i > 0$ and the loop executes. We have $b_{i+1} = a_i \mod b_i$ which is a natural number from 0 to $b_i - 1$ inclusive. Thus, $b_{i+1} < b_i$, and $b_0, b_1, ...$ is a descending sequence. Thus, the algorithm terminates!