Intro: Greatest Common Divisors I

Daniel Kane

Department of Computer Science and Engineering
University of California, San Diego

Data Structures and Algorithms
Algorithmic Toolbox

Learning Objectives

- Define greatest common divisors.
- Compute greatest common divisors inefficiently.

GCDs

- Put fraction $\frac{a}{b}$ in simplest form.
- Divide numerator and denominator by d, to get $\frac{a/d}{b/d}$.

GCDs

- Put fraction $\frac{a}{b}$ in simplest form.
- Divide numerator and denominator by d, to get $\frac{a/d}{b/d}$.
  - Need d to divide a and b.
  - Want d to be as large as possible.

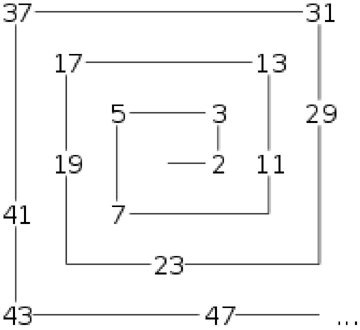
GCDs

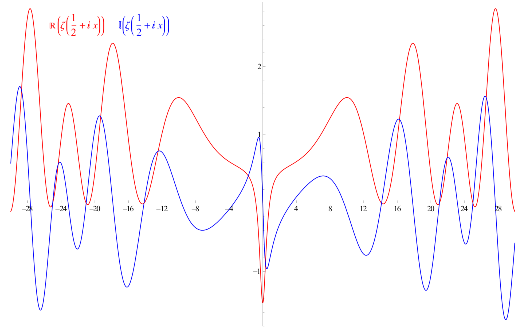
- Put fraction $\frac{a}{b}$ in simplest form.
- Divide numerator and denominator by d, to get $\frac{a/d}{b/d}$.
  - Need d to divide a and b.
  - Want d to be as large as possible.

Definition

For integers, a and b, their greatest common divisor or $\gcd(a, b)$ is the largest integer d so that d divides both a and b.

Number Theory





Cryptography

Computation

Compute GCD

Input: Integers $a, b \geq 0$.

Output: $\gcd(a, b)$.

Computation

Compute GCD

Input: Integers $a, b \geq 0$.

Output: $\gcd(a, b)$.

Run on large numbers like

$$\gcd(3918848, 1653264).$$

Naive Algorithm

```
Function NaiveGCD(a, b)
best ← 0
for d from 1 to a + b:
  if d|a and d|b:
    best ← d
return best
```

Naive Algorithm

```
Function NaiveGCD(a, b)
best ← 0
for d from 1 to a + b:
  if d|a and d|b:
    best ← d
return best
```

- Runtime approximately $a + b$.
- Very slow for 20 digit numbers.

Intro: Greatest Common Divisors II

Daniel Kane

Department of Computer Science and Engineering
University of California, San Diego

Data Structures and Algorithms
Algorithmic Toolbox

Learning Objectives

- Implement the Euclidean Algorithm.
- Approximate the runtime.

GCDs

Definition

For integers, a and b, their greatest common divisor or $\gcd(a, b)$ is the largest integer d so that d divides both a and b.

GCDs

Definition

For integers, a and b, their greatest common divisor or $\gcd(a, b)$ is the largest integer d so that d divides both a and b.

Compute GCD

Input: Integers $a, b \geq 0$.

Output: $\gcd(a, b)$.

Key Lemma

Lemma

Let a' be the remainder when a is divided by b, then

$$\gcd(a, b) = \gcd(a', b) = \gcd(b, a').$$

Proof

Proof (sketch)

- $a = a' + bq$ for some q
- d divides a and b if and only if it divides a' and b

Euclidean Algorithm

Function EuclidGCD(a, b)

```
if b = 0:
  return a
a' ← the remainder when a is
   divided by b
return EuclidGCD(b, a')
```

Euclidean Algorithm

Function EuclidGCD(a, b)

```
if b = 0:
  return a
a' ← the remainder when a is
    divided by b
return EuclidGCD(b, a')
```

Produces correct result by Lemma.

Example

$$\gcd(3918848, 1653264)$$

Example

$$
\begin{aligned}
&\gcd(3918848, 1653264) \\
= &\gcd(1653264, 612320)
\end{aligned}
$$

Example

$$
\begin{aligned}
&\gcd(3918848, 1653264)\\
=&\gcd(1653264, 612320)\\
=&\gcd(612320, 428624)
\end{aligned}
$$

Example

$$\begin{aligned} &\gcd(3918848, 1653264) \\ =\ &\gcd(1653264, 612320) \\ =\ &\gcd(612320, 428624) \\ =\ &\gcd(428624, 183696) \end{aligned}$$

Example

$$\begin{aligned} & \gcd(3918848, 1653264) \\ = & \gcd(1653264, 612320) \\ = & \gcd(612320, 428624) \\ = & \gcd(428624, 183696) \\ = & \gcd(183696, 61232) \end{aligned}$$

Example

$$
\begin{aligned}
&\gcd(3918848, 1653264)\\
=\,&\gcd(1653264, 612320)\\
=\,&\gcd(612320, 428624)\\
=\,&\gcd(428624, 183696)\\
=\,&\gcd(183696, 61232)\\
=\,&\gcd(61232, 0)
\end{aligned}
$$

Example

$$
\begin{aligned}
&\gcd(3918848, 1653264)\\
=&\gcd(1653264, 612320)\\
=&\gcd(612320, 428624)\\
=&\gcd(428624, 183696)\\
=&\gcd(183696, 61232)\\
=&\gcd(61232, 0)\\
=&61232.
\end{aligned}
$$

Runtime

- Each step reduces the size of numbers by about a factor of 2.
- Takes about $\log(ab)$ steps.

Runtime

- Each step reduces the size of numbers by about a factor of 2.
- Takes about $\log(ab)$ steps.
- GCDs of 100 digit numbers takes about 600 steps.
- Each step a single division.

Summary

- Naive algorithm is too slow.
- The correct algorithm is much better.
- Finding the correct algorithm requires knowing something interesting about the problem.