Lecture 1: Algorithmic Number Theory

Algorithmic Number Theory (Fall 2014)
Rutgers University
Swastik Kopparty
Scribe: Michael Donders

1 Course Info

- Website: http://math.rutgers.edu/ sk1233/courses/ANT-F14/
- Scribing: each person scribes 1-2 lectures, guidelines are posted on the website
- Homework: 1-2 problem sets
- Posted links to references on website
- Instructor email: swastik.kopparty@gmail.com
- Office Hours: Wed. 1:30-2:30 pm, Hill 432
- Some classes will be 5-8

2 Plan

Algorithmic Number Theory is the study of algorithms for problems involving numbers. We will use this as an excuse to see some neat ideas in number theory and in theoretical computer science.

Some of the topics we will see:

- Solving polynomial equations
- Primality testing
- Integer Factorization
- Lattices / applications of lattices
- Complexity of elementary operations
  - addition, multiplication, division, GCDs
- Polynomials Analogues of all the above
- Algorithms for finite fields
  - eg. find square roots
  - constructing finite fields

- Algorithms for elliptic curves. Elliptic curves are the curves cut out by degree 3 polynomials in 2 variables. Why should one care about them? There are some terribly deep applications of them to algorithmic problems which have nothing to do with elliptic curves:
  - Given a prime $p \equiv 1 \mod 4$, we know (by Fermat) that p can be written as $x^2 + y^2$. The only known polynomial time algorithm to find such an x, y comes by studying elliptic curves over finite fields.
  - Some of the best known error-correcting codes, the "algebraic-geometric codes" are based on understanding the space of all elliptic curves over a finite field (this won't be covered in this class).
- Algorithms for number fields.
- Hilberts 10^{th} problem.
  - Hilbert asked: Find an algorithm, which when given $Q(X_1, \ldots, X_n) \in \mathbb{Z}[X_1, \ldots, X_n]$, determines whether it has any solutions in $\mathbb{Z}^n$. Interestingly, in his formulation of the question, failure is not an option. The problem is formulated as a command: find an algorithm[1].
  - *There is no algorithm whatsoever to solve this problem!* We will see a proof in this class.
- Efficient quantum algorithms for factoring integers.
- Other assorted number theoretic algorithms.

3 GCDs and recognizing rational numbers

We will start wth the problem of computing the GCD of two numbers efficiently. Given integers $a, b > 0$,
$GCD(a, b) =$ largest $d > 0$ such that $d|a$ and $d|b$.

Before understanding the complexity of this problem, let us get warmed up and think about the complexity of our standard elementary school algorithms for basic arithmetic operations. Since a is represented using $\lceil \log_2 a \rceil$ bits, so we cant hope to do things in less than $\log a + \log b$ time.

- **Addition** Naive algorithm takes $O(\log a + \log b)$ time. (Same for subtraction.)
- **Multiplication** Naive algorithm takes $O(\log a \log b)$ time. (Same for division with remainder.)
- **Factoring** To factor a by trial division may have to go up to $\sqrt{a}$. Running time $\geq \sqrt{a} = \exp(\log a)$.

Our elementary school algorithm for computing GCDs involved factoring, which as far as we know cannot be solved in polynomial time. It turns out that one can compute GCDs in time $\texttt{poly}(\log a, \log b)$. This is one of the first nontrivial polynomial time algorithms every discovered.

[1] Hilbert also famously said: "Wir müssen wissen – wir werden wissen!", which translates to "We must know – we will know!" But again, this was before Godel and Turing.

3.1 Euclids GCD algorithms

Euclid's Observation
Euclid observed that, assuming $a > b$, $GCD(a, b) = GCD(a - b, b)$. This gives rise to Euclid's algorithm for $GCD(a, b)$:

- Compute q, r such that $a = qb + r,\ 0 \leq r < b$
- If $r = 0$, answer is b.
- Else, the answer is $GCD(b, r)$ (which we compute by recursion).

The correctness follow immediately from Euclid's observation.

3.2 Runtime of algorithm

It is trivial to see that this algorithm halts in finite time within b recursive calls.

Theorem 1. *Number of interations* $\leq O(\log b)$.

Fibonacci Numbers
$F_0 = 1$
$F_1 = 1$
$F_{n+1} = F_n + F_{n-1}$, for $n \geq 1$.

Theorem 2. *If Euclidean Algorithm for* $GCD(a, b)$ *takes* n *steps, then* $a \geq F_n$, *and* $b \geq F_{n+1}$.

Proof. Proof by induction. Clear for $n = 1$.
Suppose we have this for $n \leq N$.
Suppose $GCD(a, b)$ takes $\geq N + 1$ steps. Then $a = qb + r$. Then $GCD(r, b)$ takes $\geq N$ steps.

$$\Rightarrow b \geq F_n,\ r \geq F_{n-1}$$

$$\Rightarrow a \geq b + r = F_n + F_{n-1} = F_{N+1}$$

$$\Rightarrow a \geq F_{N+1}, b \geq F_N.$$

□

Growth of Fibonacci Numbers:
$GCD(F_N, F_{N-1}) = GCD(F_{N-1}, F_{N-2}) = \ldots$
$F_n = \Theta\left(\left(\frac{1+\sqrt{5}}{2}\right)^N\right)$.

If $a, b \leq n$-bits long then $GCD(a, b)$ will halt within $O(n)$ steps.
If a, b are $\leq n$ bits long there are $O(n)$ iterations, and each iteration takes $O(n^2)$ time (for the division), so the total time $\leq O(n^3)$.
It is known how to compute the GCD of two n bit numbers in time $O(n \cdot \log^{O(1)} n)$. We might see this in class.

3.3 Finding a rational approximation

Problem: Given $\alpha \in \mathbb{Q}$ presented as a ratio of n bit integers, and $Q \in \mathbb{N}$, find $a, q \in \mathbb{N}$ such that

$$|\alpha - \frac{a}{q}| = \min_{a,q;q \leq Q} |\alpha - \frac{a}{q}|.$$

That is the closest integer fraction to α with bounded denominator.
Hope: Runtime $O(n + logQ)$ or $(n + logQ)^{O(1)}$. Note that it is trivial to get $O(Q + n)$ time (By trying all denominators).

3.4 Continued Fractions

Consider the execution of Euclid's GCD algorithm on a, b:

$$a = bq_0 + r_0$$

$$b = r_0 q_1 + r_1$$

$$r_0 = r_1 q_2 + r_2$$

$$\vdots$$

$$r_{n-2} = r_{n-1} q_n + 0$$

This can be interpreted as a certain kind of representation of the rational number a/b:

$$\frac{a}{b} = q_0 + \frac{r_0}{b}$$

$$\frac{b}{r_0} = q_1 + \frac{r_1}{r_0}$$

$$\frac{r_0}{r_1} = q_2 + \frac{r_2}{r_1}$$

$$\vdots$$

$$\frac{r_{n-2}}{r_{n-1}} = q_n$$

This representation is called the continued fraction representation (which can be defined for any real number), and is closely related to the problem of finding a good rational approximation to a given real number.

$$\frac{a}{b} = q_0 + \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 + \ddots \cfrac{1}{q_n}}}} \tag{1}$$

Continued fraction representation.
Given $\alpha \in \mathbb{R}^+$. $\alpha = \lfloor \alpha \rfloor + \frac{1}{\alpha_1}$, $1 < \alpha_1 < \infty$
$\alpha_1 = \lfloor \alpha_1 \rfloor + \frac{1}{\alpha_2}$

$\alpha_2 = \lfloor \alpha_2 \rfloor + \frac{1}{\alpha_3}$
Labeling $\lfloor \alpha_i \rfloor = a_i$, we have that

$$\alpha = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \ddots}}} \tag{2}$$

Turning each iteration into an iteration fraction
$\frac{p_0}{q_0} = a_0$; $\frac{p_1}{q_1} = a_0 + \frac{1}{a_1}$; $\frac{p_2}{q_2} = a_0 + \frac{1}{a_1 + \frac{1}{a_2}}$; $\dots$
Introducting a formal variable z we can write α as

$$\alpha = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \ddots \cfrac{1}{a_n + \cfrac{1}{z}}}}} \tag{3}$$

Simplifying the right hand expression for z we get $\frac{rz+s}{tz+u}$. By evaluating at specific points we can see,
$z = \infty \to \frac{p_n}{q_n}$
$z = a_{n+1} \to \frac{p_{n+1}}{q_{n+1}}$
$z = 0 \to \frac{p_{n-1}}{q_{n-1}}$
From here this implies

$$\frac{rz+s}{tz+u} = \frac{p_n z + p_{n-1}}{q_n z + q_{n-1}}.$$

To prove this we note

$$\begin{bmatrix} p_n & p_{n-1} \\ q_n & q_{n-1} \end{bmatrix} \begin{bmatrix} a_{n+1} & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} p_{n+1} & p_n \\ q_{n+1} & q_n \end{bmatrix}$$

$$\begin{bmatrix} p_1 & p_0 \\ q_1 & q_0 \end{bmatrix} := \begin{bmatrix} a_1 a_0 + 1 & a_0 \\ a_1 & 1 \end{bmatrix}$$

$$\det \begin{pmatrix} p_n & p_{n-1} \\ q_n & q_{n-1} \end{pmatrix} = (-1)^{n+1}$$

This implies $pnq_{n-1} - qnp_{n-1} = \pm 1$, which in turn implies p_n, q_n are relatively prime, as well as p_n, p_{n-1} and q_n, q_{n-1} are relatively prime.

In the next class, we will prove some properties of continued fractions, and see how to use continued fractions to solve the rational approximation problem.