Today: Division - compute leading bits of \( x/y \)

Elementary-school approach: \[ \frac{0.10101}{11.0000} \]

Simplifications
1) Focus on computing \( x/y \) because can mult by \( x \)
2) Rescale \( y \) so that \( 1/2 \leq y < 1 \Rightarrow 1 \leq xy < 2 \)

First Approaches: Binary Search

Let \( x_i \) = \( i \)th guess for \( xy \)

\[ x_0 = \frac{3}{2} \]

\[ \frac{1}{2} \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \]

Ex: \( y = 1/3 \Rightarrow y = 3/4 \)

Performance: One bit of accuracy per iteration

\( O(n) \) rounds \( \Rightarrow O(n \log n) \) time.
Faster approach: Newton Iteration

Goal: approx a root of $f(x)$.

$$f'(x_i) = \frac{f(x_i)}{\Delta}$$

$$X_{i+1} = X_i - \frac{f(x_i)}{f'(x_i)}$$

To compute $y$, find root of

$$f(x) = 1 - xy$$

$$\Rightarrow f'(x) = -y.$$
\[ y = 0.11 \]

\[ x_0 = 1.1 \]
\[ x_1 = 1.0101 \]
\[ x_2 = 1.0010101001 \]
\[ x_3 \]
Division on $N$-cell Linear Array

$LgN$ iterations each composed of $O(N)$ steps.

$\Rightarrow O(NlgN)$ steps on $N$-cell linear array

Better idea:

Precision of $x_i$ only kept to $2^{i+1}$ bits.

Cost: $T(n) = T(n/2) + O(n) = O(n)$. 
Computing \( y/\ln y \) in \( O(\log n) \) steps

Easy case: \( y \) fixed \( \Rightarrow \) precompute \( y \).

Simplifications:

1) Focus on \( y \) (as before)

2) Rescale so \( y = 1 - x \), \( 0 \leq x \leq 1/2 \) (as before)

3) \( y = 1 - x \)

\[ = 1 + x + x^2 + x^3 + \cdots + \]

Let

\[ x_i = 1 + x + x^2 + \cdots + x^i \]

\[ |y - x_i| = x^{i+1} + x^{i+2} + \cdots \]

\[ \leq \frac{1}{2^{i+1}} + \frac{1}{2^{i+2}} + \cdots \]

\[ \leq \frac{1}{2^i} \]

\( \Rightarrow \) Sufficient to compute \( x_i \)
Need $N + \lg N$ bits

\[ \frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \cdots + \frac{1}{2^N} \]

Wallace Tree

Need $N$ bits

Reduce to calculating $\frac{1}{2^i}$, $i = 0 \ldots N$.  

Naïve: Repeated squaring $\Rightarrow \Theta(\lg^2 N)$.  

$\Theta(\lg N)$
Chinese Remainder Theorem

Let $p_1, p_2, \ldots, p_s$ be prime numbers.
Let $P = p_1 p_2 \ldots p_s$.

For any number $Z$, define the vector of residues to be $(z_1, z_2, \ldots, z_s)$, where $0 \leq i < p_i$ and $z_i = Z \mod p_i$ ($i = 1 \ldots s$).

For each $Z$, $0 \leq Z < P$, the vector of residues is unique. Moreover the value of $Z$ can be calculated from its residues by setting

$$Z = \sum_{i=1}^{s} \beta_i z_i \mod P,$$

where

$$\beta_i = \left(\frac{P}{p_i}\right) d_i$$

and

$$d_i = \left(\frac{P}{p_i}\right)^{-1} \mod p_i.$$

Represent numbers with CRT encoding

$$Z \leftrightarrow (z_1, \ldots, z_s)$$
Example of CRT:

$p_1 = 2$
$p_2 = 3$
$p_3 = 5$
$p_4 = 7$

$P = p_1 p_2 p_3 p_4 = 210.$

$\alpha_1 \equiv \left( \frac{210}{2} \right)^{-1} \equiv (105)^{-1} \equiv 1 \pmod{2}$

$\alpha_2 \equiv \left( \frac{210}{3} \right)^{-1} \equiv (70)^{-1} \equiv 1 \pmod{3}$

$\alpha_3 \equiv \left( \frac{210}{5} \right)^{-1} \equiv (42)^{-1} \equiv 3 \pmod{5}$

$\alpha_4 \equiv \left( \frac{210}{7} \right)^{-1} \equiv (30)^{-1} \equiv 4 \pmod{7}$

$\beta_1 = (\frac{210}{2}) \cdot 1 = 105$

$\beta_2 = (\frac{210}{3}) \cdot 1 = 70$

$\beta_3 = (\frac{210}{5}) \cdot 3 = 126$

$\beta_4 = (\frac{210}{7}) \cdot 4 = 120$
Ex. For any $P = \frac{2}{3} \pi + \frac{7}{2}$, can represent any $Z \leq 210$.

$\beta_1 = 105$
$\beta_2 = 70$
$\beta_3 = 126$
$\beta_4 = 120$

If $Z = 132$, $(\beta_1, \beta_2, \beta_3, \beta_4) = (105, 70, 126, 120)$

$132 = 0 \cdot 105 + 0 \cdot 70 + 2 \cdot 126 + 6 \cdot 120 \mod 210$

$42 \equiv 80$

If $Z = 70$, $(\beta_1, \beta_2, \beta_3, \beta_4) = (0, 1, 0, 0)$

$70 = 1 \cdot 70$
Computing $Z$ from $(z_1, z_2, \ldots, z_s)$.

$$Z = \sum_{i=1}^{s} \beta_i z_i \mod p$$

Taking mods:

$$\text{blah \mod p} = \text{blah} - \left\lfloor \frac{\text{blah}}{p} \right\rfloor \cdot p$$

$\frac{p}{p}$ precomputed
Computing $\mathbb{Z}^N$ in CRT Notation

<< Need $P$ big enough to represent $\mathbb{Z}^N$ >>

Need $P = \mathbb{Z}^N$

$> (\mathbb{Z}_p)^N$

$> \mathbb{Z}^{N^2}$

Sufficient that $P = p_1 p_2 \ldots p_{N^2}$.

$\mathbb{Z}^N = \sum_{i=1}^{N^2} \beta_i (\mathbb{Z}^N \mod p_i) \mod P$

Calculated by computing

$g_i^N \quad (1 \leq i \leq s)$

$(\mathbb{Z} \mod p_i)^N$

Lemma: Each $g_i (1 \leq i \leq s)$ represented with $\Theta(\log N)$ bits.

<< In contrast, $\mathbb{Z}$ represented with $\Theta(N)$ bits. >>

Pf: By Prime Number Theorem, which says

#primes $< N$ is $\Theta(N/\log N)$.

$\Rightarrow$ our largest prime only $\Theta(N^2/\log N)$. 
\[ Z^N = \sum_{i=1}^{N^2} \beta_i (Z^N \mod P_i) \mod P \]

Big Question: How to compute \( Z^N \mod p_i \)?

Good Answer: Since only \( \Theta(lg N) \) bits \( \Rightarrow \Theta(lg N \ lg lg N) \) time.

<< But can do better! >>

Better: Lookup Tables!!!
Summary

\[ \frac{1}{y} = \frac{1}{1 - z} \]

\[ (1) + (Z) + (Z^2) + \ldots + (Z^N) \]

\[ \sum_{i=1}^{N^2} \beta_i Z^i \mod P \]

Wallace Tree