Michael Bender lecturing

6.896
2/17/04
L4

Today: Division — compute n leading bits of u/y.

Elementary-school approach: $1/3$

```
     .010101
11 ) 1.00000
      11
      ----
      100
       11
      ----
       100
```

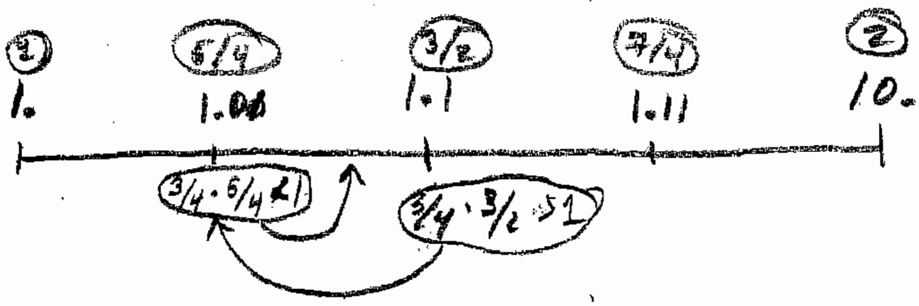
Simplifications

1) Focus on computing $1/y$ because can mult by u
2) Rescale y so that $1/2 \le y < 1 \Rightarrow 1 < 1/y \le 2$.

First Approach: Binary Search

Let x_i = ith guess for $1/y$

$x_0 = 3/2$.



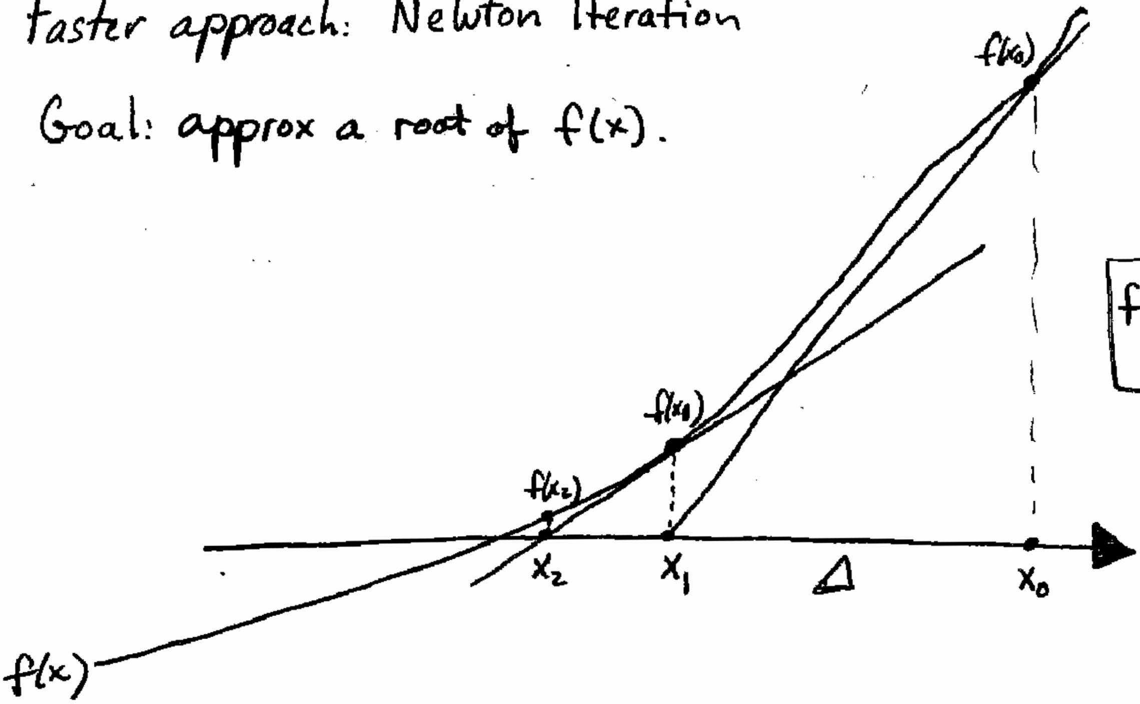
Ex: $y = 13 \Rightarrow y = 3/4$

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 $f(x)$.



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

$$\boxed{x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}}$$

To compute $1/y$, find root of

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

$$\Rightarrow \quad f'(x) = -y.$$

$$x_{i+1} = x_i - \frac{1 - x_i y}{-y}$$

$$= x_i + \frac{1}{y}(1 - x_i y)$$

<< Uh oh. To compute $\frac{1}{y}$, all we need is $1/y$. >>

Ex: $y = .11$

$x_0 = 1.1$

$x_1 = 1.0101$

$x_2 = 1.0101010001$

etc

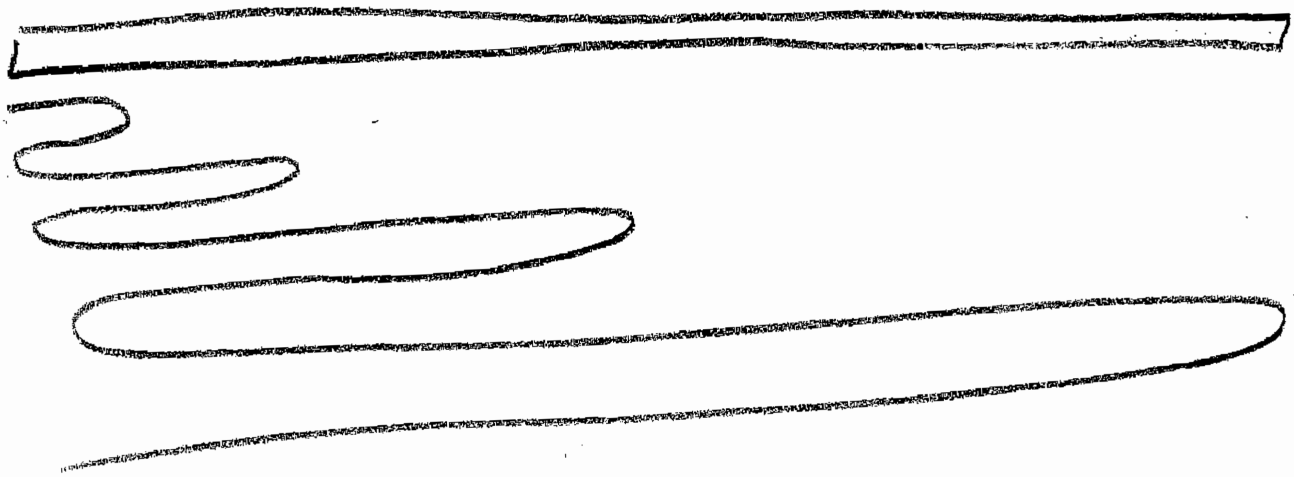
Division on N-cell Linear Array

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

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

<u>Better idea:</u>

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



<u>Cost</u>: $T(N) = T(N/2) + \Theta(N)$

$= \Theta(N)$.

Computing u/y in $O(\lg n)$ steps

Easy case: y fixed $\Rightarrow$ precompute $1/y$.

Simplifications:

1) focus on $1/y$ (as before)

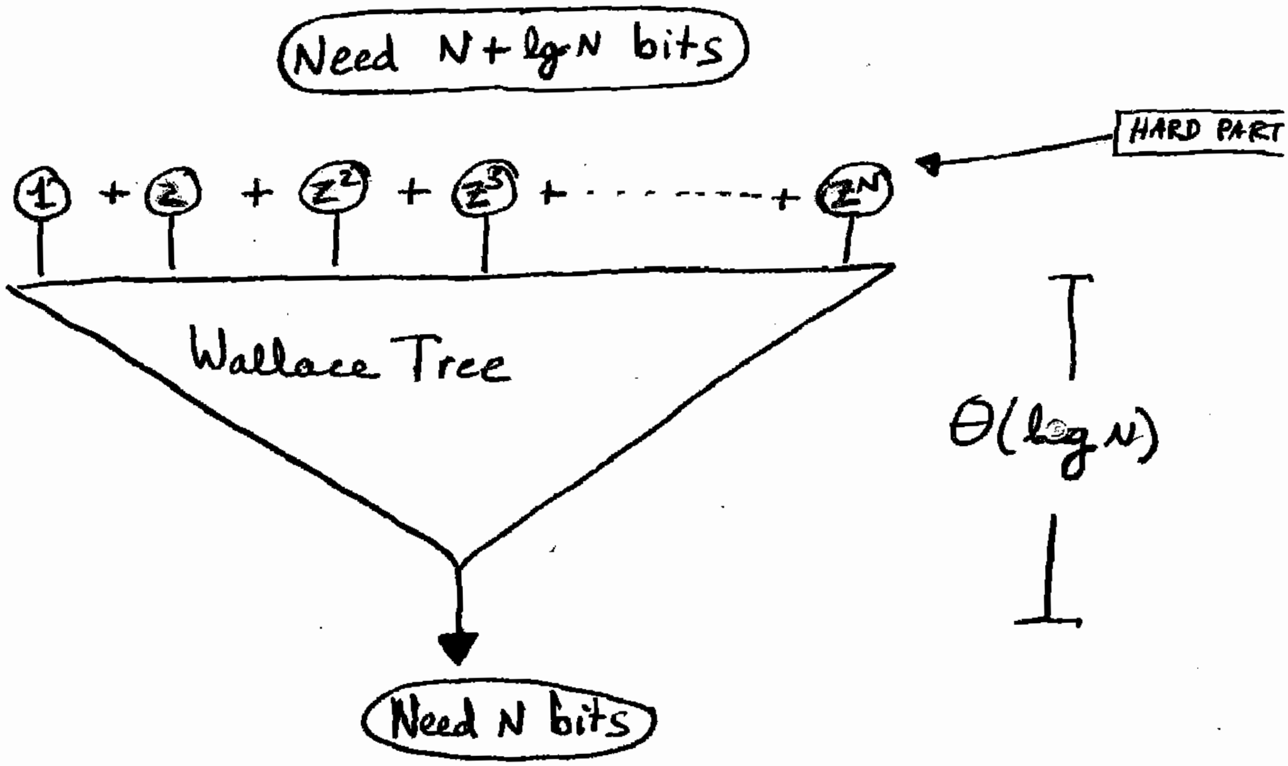
2) rescale so $y = 1 - z$, $0 \le z \le 1/2$ (as before)

3) $$\frac{1}{y} = \frac{1}{1-z}$$
$$= 1 + z + z^2 + z^3 + \cdots +$$

Let
$$x_i = 1 + z + z^2 + \cdots + z^i$$

$$|1/y - x_i| = z^{i+1} + z^{i+2} + \cdots$$
$$\le \frac{1}{2^{i+1}} + \frac{1}{2^{i+2}} + \cdots$$
$$\le \frac{1}{2^i}$$

$\Rightarrow$ sufficient to compute x_n



Reduce to calculating z^i, $i = 0 \ldots N$.

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

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 $0 \le i < p_i$ and $z_i = Z \bmod p_i$ $(i = 1 \ldots s)$.

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

precomputed for all Z

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

where

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

and

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

Represent numbers with CRT encoding

$$Z \longleftrightarrow (z_1, \ldots, z_s)$$

Example of CRT:

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

$P = p_1 p_2 p_3 p_4 = 210.$

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

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

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

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

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

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

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

$$\beta_4 = \left(\frac{210}{7}\right) \cdot 4 = 120$$

Ex For any $P = 2 \cdot 3 \cdot 5 \cdot 7$, can represent any $Z < 210$.
$= 210$

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

If $Z = 132$, $(z_1, z_2, z_3, z_4) = (0, 0, 2, 6)$

$$132 = 0 \cdot 105 + 0 \cdot 70 + \underbrace{2 \cdot 126}_{42} + \underbrace{6 \cdot 120}_{90} \bmod 210$$

If $Z = 70$, $(z_1, z_2, z_3, z_4) = (0, 1, 0, 0)$

$$70 = \qquad 1 \cdot 70$$

Computing z from $(z_1, z_2, \ldots z_s)$.

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

Taking mods:

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

$\boxed{\frac{1}{p} \text{ precomputed}}$

Computing Z^N in CRT Notation

<< Need P big enough to represent Z^N >>

Need $P > Z^N$

$> (2^N)^N$

$> 2^{N^2}$

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

$$Z^N = \sum_{i=1}^{N^2} \beta_i \left(Z^N \bmod p_i \right) \bmod P$$

Calculated by computing

$z_i^N \quad (1 \le i \le s)$

$\shortparallel$

$(Z \bmod p_i)^N$

Lemma: Each z_i $(1 \le i \le s)$ represented with $\Theta(\lg N)$ bits.

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

Pf. By Prime Number Theorem, which says

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

$\Rightarrow$ our largest prime only $\Theta(N^2 \lg N)$.

$$Z^N = \sum_{i=1}^{N^2} \beta_i \left(Z^N \bmod P_i \right) \bmod P$$

precomputed

$1/P_i$ precomputed

$1/P$ precomputed

Big Question: How to compute $Z^N \bmod p_i$??

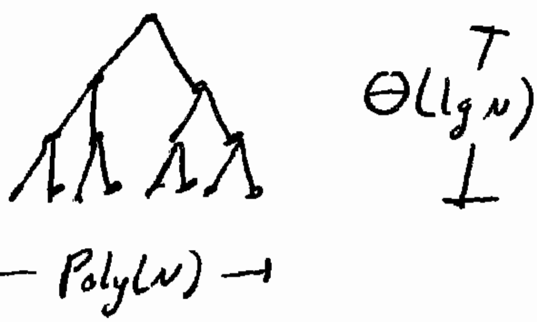
Good Answer: Since only $\Theta(\lg N)$ bits $\Rightarrow$ $\Theta(\lg N \lg\lg N)$ time.
<< But can do better! >>

$\boxed{\text{Better: Lookup Tables!!!}}$

$\forall\ p_i,\ Z \bmod p_i,$ precompute $(Z^N \bmod p_i)$.

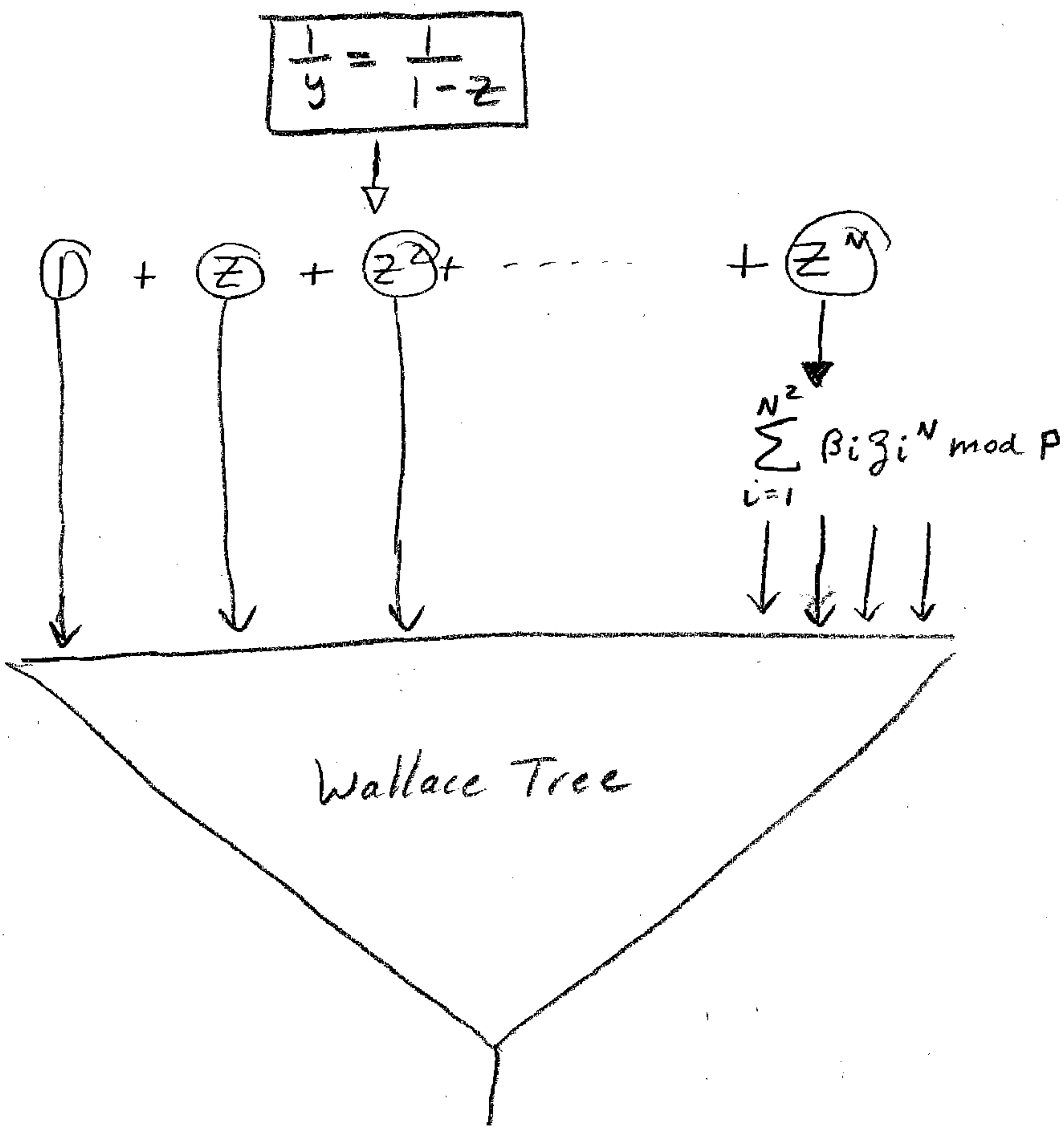
N^2 choices

$2^{\Theta(\lg N)}$ choices



$\Rightarrow O(\lg N)$ time per lookup.

Summary


$\boxed{\frac{1}{y} = \frac{1}{1-z}}$
$1 + z + z^2 + \cdots + z^N$
$\sum_{i=1}^{N^2} \beta_i z_i^N \bmod p$
Wallace Tree