CS 681: Computational Number Theory and Algebra
Lecture 4: Arithmetic over Z:Division
Lecturer: Manindra Agrawal **Notes by: Barna Saha**

August 11, 2006.

1 Introduction

Given two n-bit numbers a and b, we know addition, that is computing $a+b$ takes $O(n)$ time. If we want to multiply a and b, the naive algorithm will take $O(n^2)$ time. However the time complexity of multiplying 2, n bit numbers can be brought down to $O(n \log n \log\log n)$. It is easy to see, that we can perform division of a by b, that is compute $\frac{a}{b}$, in $O(n^2)$ time. Since computing $\frac{a}{b}$ is nothing but multiplying a by reciprocal of b, the natural question arises, whether it is possible to achieve the same time complexity for division as that of multiplication. In this lecture, we try to find out an answer for this. ¿From here on, we denote the time complexity of multiplying two n bit numbers by $M(n)$.

2 Newton Approximation/ Iteration & Division

2.1 Newton Approximation/Iteration

Newton Approximation/Iteration method is used to compute approximate root of any function $f(x)$ iteratively. It starts from an arbitrary point x_0 and on each iteration it calculates a new value-a new approximation of the root, based on the value computed at the previous iteration. It keeps on continuing, until the difference of the values computed in consecutive iterations is below a threshold. If we denote the value computed at any ith iteration by x_i, then x_{i+1} is computed as,

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

Here $f'(x_i)$ represents derivative of $f(x)$ at x_i.

2.2 Division

Problem Definition. Given integers a, b and k, compute $\frac{a}{b}$ upto k bits of precision.

Observation. We can assume $a = 1$. For any other a, we simply need to mutiply that with $\frac{1}{b}$. Let $|b| = n$.

Consider the function $f(x) = \frac{1}{x} - b$. Clearly, the root of $f(x)$, gives the value of $\frac{1}{b}$. We use Newton Approximation method, to find the root of $f(x)$, upto k bits of precision. Since

$f(x) = \frac{1}{x} - b$, we have $f'(x) = -\frac{1}{x^2}$. Therefore by equation 1, we get

$$\begin{aligned} x_{i+1} &= x_i - \frac{\frac{1}{x_i} - b}{-\frac{1}{x_i^2}} \\ &= 2x_i - bx_i^2 = x_i(2 - bx_i) \qquad (2) \end{aligned}$$

Now we need to show, that x_i's converge towards $\frac{1}{b}$, as iteration proceeds. If this holds, then we can stop iterating, after the required precision has been obtained and output the result. The following lemma establishes this convergence result.

Lemma 2.1 *Assume* $b \geq 3$, $\left|x_i - \frac{1}{b}\right| \leq \frac{1}{b 2^{2^i}}$

Proof: Proof by Induction:

[i=0] If $2^m \leq b \leq 2^{m+1}$, start with $x_0 = \frac{1}{2^m}$.

[Induction Step] Suppose the result is true for i.

Now $\left|x_i - \frac{1}{b}\right| \leq \frac{1}{b 2^{2^i}}$ iff $\left(x_i - \frac{1}{b}\right)^2 \leq \frac{1}{b^2 2^{2^{i+1}}}$.

Consider for $i + 1$.

$$\begin{aligned} \left(x_{i+1} - \frac{1}{b}\right)^2 &= \left(2x_i - bx_i^2 - \frac{1}{b}\right)^2 \\ &= \left(\frac{2bx_i - b^2x_i^2 - 1}{b}\right)^2 \\ &= \frac{(bx_i - 1)^4}{b^2} \\ &= b^2\left(x_i - \frac{1}{b}\right)^4 \\ &\leq \frac{1}{b^2 2^{2^{i+2}}} \qquad (3) \end{aligned}$$

Hence $\left|x_{i+1} - \frac{1}{b}\right| \leq \frac{1}{b 2^{2^{i+1}}}$. ■

Time Complexity.

1. In each iteration we need to perform 2 multiplications and one addition. Hence time required for each iteration is $O(M(n))$.

2. If $k = n$, total number of iterations required is $\leq \log n$.

Therefore the total time compexity is $O(M(n) \log n)$.

However this extra $\log n$ factor can be removed. Noticing that the time of ith iteration is in fact $O(M(2^i))$ and $M(a)+M(b) \leq M(a+b)$, we will achieve $O(M(n))$ time complexity for division. Filling the gap, is given as part of Assignment 1.