Lecture 11: Numerics I

Lecture Overview

- Irrationals
- Newton's Method ($\sqrt{(a)}, 1/b$)
- High precision multiply ←

Irrationals:

Pythagoras discovered that a square's diagonal and its side are incommensurable, i.e., could not be expressed as a ratio - he called the ratio "speechless"!

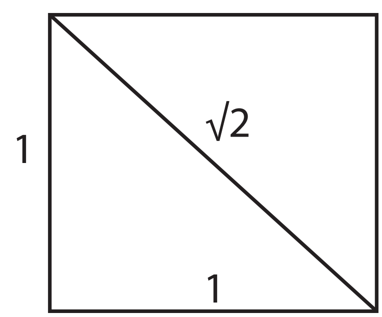

Figure 1: Ratio of a Square's Diagonal to its Sides.

Pythagoras worshipped numbers
"All is number"
Irrationals were a threat!

Motivating Question: Are there hidden patterns in irrationals?

$$\sqrt{2} = 1.\ 414\ 213\ 562\ 373\ 095\ 048\ 801\ 688\ 724\ 209\ 698\ 078\ 569\ 671\ 875$$

Can you see a pattern?

Digression

Catalan numbers:
Set P of <u>balanced</u> parentheses strings are recursively defined as

- $\lambda \in P \quad (\lambda$ is empty string)
- If $\alpha, \beta \in P$, then $(\alpha)\beta \in P$

Every nonempty balanced paren string can be obtained via Rule 2 from a unique α, β pair.
For example, (()) ()() obtained by $(\underbrace{()}_{\alpha})\underbrace{()()}_{\beta}$

Enumeration

C_n: number of balanced parentheses strings with exactly n pairs of parentheses
$C_0 = 1$ empty string

C_{n+1}? Every string with $n+1$ pairs of parentheses can be obtained in a unique way via rule 2.

One paren pair comes explicitly from the rule.
k pairs from α, $n-k$ pairs from β

$$C_{n+1} = \sum_{k=0}^{n} C_k \cdot C_{n-k} \quad n \geq 0$$

$$C_0 = 1 \quad C_1 = {C_0}^2 = 1 \quad C_2 = C_0C_1 + C_1C_0 = 2 \quad C_3 = \cdots = 5$$

1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, 58786, 208012, 742900, 2674440, 9694845, 35357670, 129644790, 477638700, 1767263190, 6564120420, 24466267020, 91482563640, 343059613650, 1289904147324, 4861946401452, 18367353072152, 69533550916004, 263747951750360, 1002242216651368

Newton's Method

Find root of $f(x) = 0$ through successive approximation e.g., $f(x) = x^2 - a$

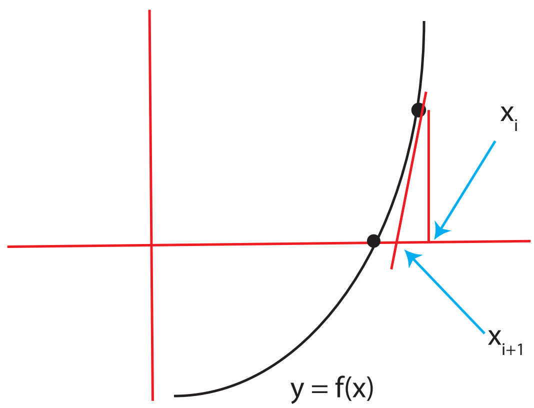

Figure 2: Newton's Method.

Tangent at $(x_i, f(x_i))$ is line $y = f(x_i) + \underline{f'(x_i)} \cdot (x - x_i)$ where $f'(x_i)$ is the derivative.
x_{i+1} = intercept on x-axis

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

Square Roots

$$f(x) = x^2 - a$$

$$\chi_{i+1} = \chi_i - \frac{({\chi_i}^2 - a)}{2\chi_i} = \frac{\chi_i + \dfrac{a}{\chi_i}}{2}$$

Example

$$\begin{aligned} \chi_0 &= 1.000000000 \qquad a = 2 \\ \chi_1 &= 1.500000000 \\ \chi_2 &= 1.416666666 \\ \chi_3 &= 1.414215686 \\ \chi_4 &= 1.414213562 \end{aligned}$$

Quadratic convergence, ♯ digits doubles. Of course, in order to use Newton's method, we need high-precision division. We'll start with multiplication and cover division in Lecture 12.

High Precision Computation

$\sqrt{2}$ to d-digit precision: $\underbrace{1.414213562373}_{\text{d digits}}\cdots$

Want integer $\lfloor 10^d\sqrt{2}\rfloor = \lfloor\sqrt{2\cdot 10^{2d}}\rfloor$ - integral part of square root

Can still use Newton's Method.

High Precision Multiplication

Multiplying two n-digit numbers (radix $r = 2, 10$)

$0 \leq x, y < r^n$

$$\begin{aligned} x &= x_1 \cdot r^{n/2} + x_0 \qquad x_1 = \text{ high half} \\ y &= y_1 \cdot r^{n/2} + y_0 \qquad x_0 = \text{ low half} \\ 0 &\leq x_0, x_1 < r^{n/2} \\ 0 &\leq y_0, y_1 < r^{n/2} \end{aligned}$$

$$z = x \cdot y = x_1 y_1 \cdot r^n + (x_0 \cdot y_1 + x_1 \cdot y_0) r^{n/2} + x_0 \cdot y_0$$

4 multiplications of half-sized ♯'s $\implies$ quadratic algorithm $\theta(n^2)$ time

Karatsuba's Method

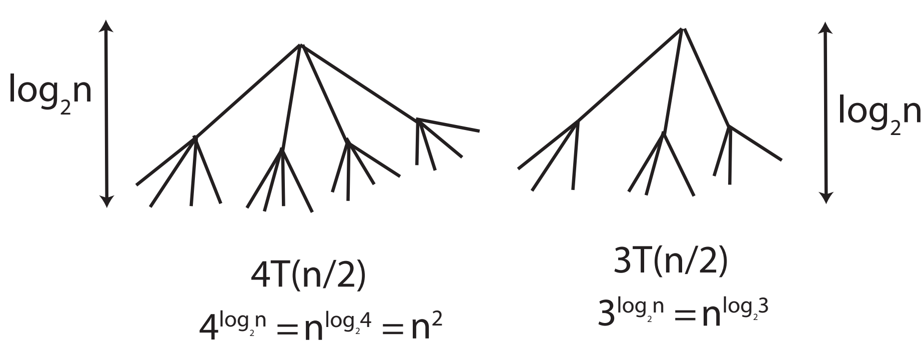

Figure 3: Branching Factors.

Let

$$\begin{aligned} z_0 &= \underline{x_0 \cdot y_0} \\ z_2 &= x_1 \cdot y_1 \\ z_1 &= (x_0 + x_1) \cdot (y_0 + y_1) - z_0 - z_2 \\ &= x_0 y_1 + x_1 y_0 \\ z &= z_2 \cdot r^n + z_1 \cdot r^{n/2} + z_0 \end{aligned}$$

There are three multiplies in the above calculations.

$$\begin{aligned} T(n) &= \text{ time to multiply two } n\text{-digit}\sharp' s \\ &= 3T(n/2) + \theta(n) \\ &= \theta\left(n^{\log_2 3}\right) = \theta\left(n^{1.5849625\cdots}\right) \end{aligned}$$

This is better than $\theta(n^2)$. Python does this, and more (see Lecture 12).

Fun Geometry Problem

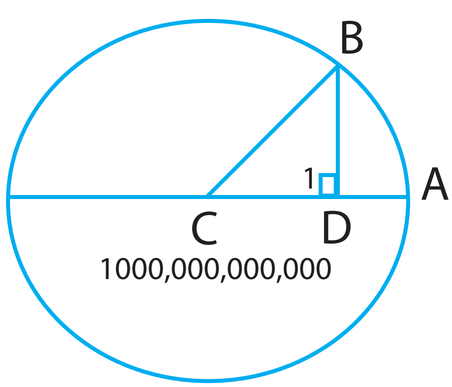

Figure 4: Geometry Problem.

$BD = 1$
What is AD?

$$AD = AC - CD = 500,000,000,000 - \sqrt{\underbrace{500,000,000,000^2 - 1}_{a}}$$

Let's calculate AD to a million places. (This assumes we have high-precision division, which we will cover in Lecture 12.) Remarkably, if we evaluate the length

to several hundred digits of precision using Newton's method, the Catalan numbers come marching out! Try it at:
`http://people.csail.mit.edu/devadas/numerics_demo/chord.html`.

An Explanation

This was *not* covered in lecture and will *not* be on a test. Let's start by looking at the power series of a real-valued function Q.

$$Q(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + \dots \tag{1}$$

Then, by ordinary algebra, we have:

$$1 + xQ(x)^2 = 1 + c_0^2 x + (c_0 c_1 + c_1 c_0)x^2 + (c_0 c_2 + c_1 c_1 + c_2 c_0)x^3 + \dots \tag{2}$$

Now consider the equation:

$$Q(x) = 1 + xQ(x)^2 \tag{3}$$

For this equation to hold, the power series of $Q(x)$ must equal the power series of $1+xQ(x)^2$. This happens only if all the coefficients of the two power series are equal; that is, if:

$$\begin{aligned} c_0 &= 1 && (4)\\ c_1 &= c_0^2 && (5)\\ c_2 &= c_0 c_1 + c_1 c_0 && (6)\\ c_3 &= c_0 c_2 + c_1 c_1 + c_2 c_0 && (7)\\ &\text{etc.} && (8) \end{aligned}$$

In other words, the coefficients of the function Q must be the Catalan numbers!

We can solve for Q using the quadratic equation:

$$Q(x) = \frac{1 \pm \sqrt{1-4x}}{2x} \tag{9}$$

Let's use the negative square root. From this formula for Q, we find:

$$10^{-12} \cdot Q(10^{-24}) = 10^{-12} \cdot \frac{1 \pm \sqrt{1 - 4 \cdot 10^{-24}}}{2 \cdot 10^{-24}} \tag{10}$$

$$= 500000000000 - \sqrt{500000000000^2 - 1} \tag{11}$$

From the original power-series expression for Q, we find:

$$10^{-12} \cdot Q(10^{-24}) = c_0 10^{-12} + c_1 10^{-36} + c_2 10^{-60} + c_3 10^{-84} + \dots \tag{12}$$

Therefore, $500000000000 - \sqrt{500000000000^2 - 1}$ should contain a Catalan number in every twenty-fourth position, which is what we observed.

MIT OpenCourseWare
http://ocw.mit.edu

6.006 Introduction to Algorithms
Fall 2011

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.