Lecture 22: Numerics I

Lecture Overview

- Irrationals
- Newton's Method ($\sqrt{(a)}, 1/b$)
- High precision multiply $\leftarrow$
- Next time
  - High precision radix conversion (printing)
  - High precision division

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"!

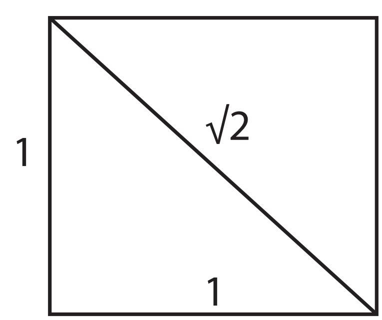

Figure 1: Ratio of a Square's Diagonal to its Sides.

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

Motivating Question: Are there hidden patterns in irrationals? Can you see a pattern?

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

Digression

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

- $\lambda \in P$ (λ 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, . . .

Geometry Problem

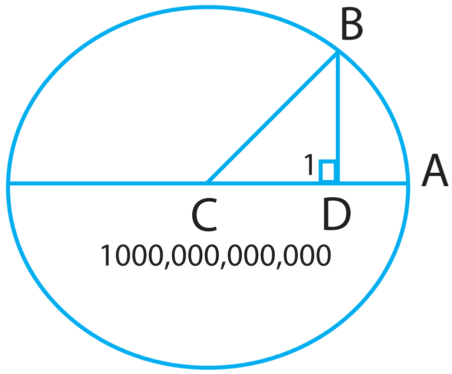

Figure 2: Geometry Problem.

$BD = 1$
What is AD?

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

Let's calculate AD to a million places!

Newton's Method

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

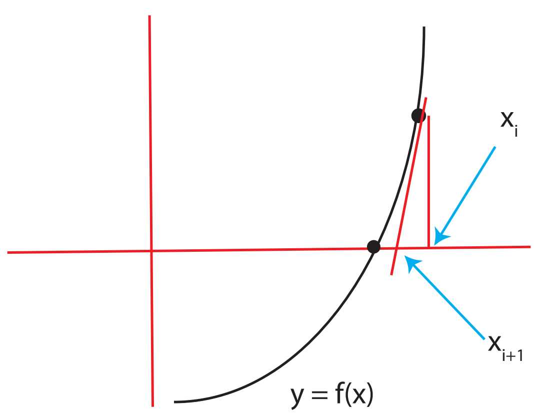

Figure 3: Newton's Method.

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

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

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

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

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.
Let's try it on $\sqrt{2}$, and our segment AD!
See anything interesting?

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

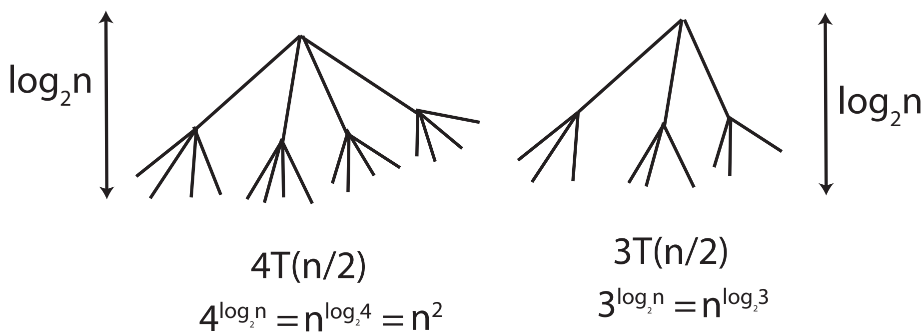

Figure 4: Branching Factors.

Let

$$
\begin{aligned}
z_0 &= x_0 \cdot y_0 \\
z_2 &= x_2 \cdot y_2 \\
z_1 &= (x_0 + x_1) \cdot (y_0 + y_1) - z_0 - z_2 \\
&= x_0 y_1 + x_1 y_0 \\
z &= z_2 \cdot r^n + z \cdot r^{n/2} + z_0
\end{aligned}
$$

There are three multiplies in the above calculations.

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

Better than $\theta(n^2)$. Python does this.