UNIT 4: NUMERICS

L11. INTEGER ARITHMETIC, KARATSUBA MULTIPLICATION

IRRATIONALS

Pythagoras worshipped numbers

"All is number"

Irrationals were a threat!

DIGRESSION

Catalan numbers:

Set P of balanced parentheses strings are recursively defined as

- $\lambda \in P \quad$ (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

k pairs from $\alpha, n-k$ pairs from β

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

$$C_0=1 \quad C_1=C_0^2=1 \quad C_2=C_0 C_1+C_1 C_0=2 \quad C_3=\cdots=5$$

NEWTON´S METHOD(IRRATIONALS)

$$x_{i+1}=x_i-\frac{f\left(x_i\right)}{f^{\prime}\left(x_i\right)}$$

Square Roots

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

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

Example

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

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 $\lfloor 10^d\sqrt{2}\rfloor = \left\lfloor\sqrt{2\cdot 10^{2d}}\right\rfloor$ – integral part of square root Can still use Newton's Method.

HIGH PRECISION MULTIPLICATION

Multiplying two n -digit numbers $(\text{radix}\, r = 2, 10)$:
$0 \leq x, y < r^n$

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

4 multiplications of half-sized #'s $\Longrightarrow$ **quadratic algorithm $\Theta\left(n^2\right)$ time**

KARATSUBA'S METHOD

Let

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

There are three multiplies in the above calculations.

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

FUN GEOMETRY PROBLEM

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

If we evaluate the length to several hundred digits of precision using Newton's method, **<u>the Catalan numbers come marching out</u>**! Try it at:

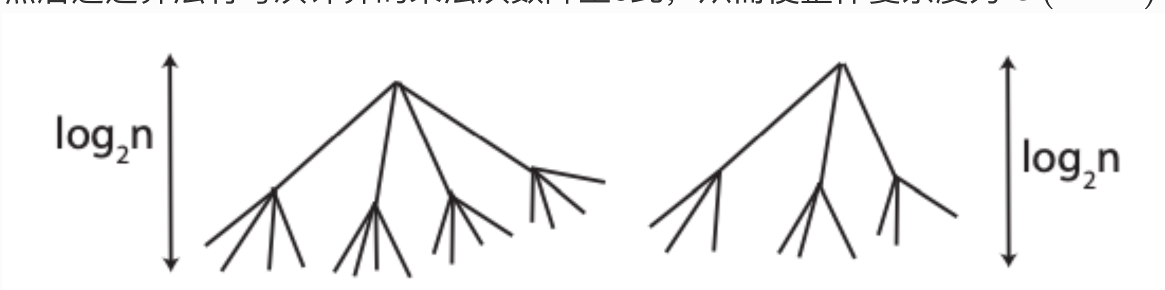
http://people.csail.mit.edu/devadas/numerics_demo/chord.html

An Explanation

see Lecture Note (Fun!)

THINKING

本讲从无理数的计算引出了大数乘法（作为大树除法的基础）。一般乘法的复杂度为 $O(n^2)$，但由于经常使用，希望有复杂度更低的算法。

- 首先通过算法将大数的位数减半，迭代或递归至word size（字长）
- 然后通过算法将每次计算的乘法次数降至3此，从而使整体复杂度为 $O(2^{\log_2 3})$
- 

$$\mathrm{4\ T(n/2)} \qquad \mathrm{3\ T(n/2)}$$
$$4^{\log_2 n} = \mathrm{n}^{\log_1 4} = \mathrm{n}^2 \quad 3^{\log_2 n} = \mathrm{n}^{\log_2 3}$$

本讲还介绍了Catalan Number，作为一个趣味知识

12. SQUARE ROOTS, NEWTON'S METHOD

ERROR ANALYSIS OF NEWTON´S METHOD

Suppose $X_n = \sqrt{a} \cdot (1 + \epsilon_n)$ $\quad \epsilon_n$ may be $+$ or $-$

Then,

$$\begin{aligned} X_{n+1} &= \frac{X_n + a/X_n}{2} \\ &= \frac{\sqrt{a}\,(1+\epsilon_n) + \frac{a}{\sqrt{a}(1+\epsilon_n)}}{2} \\ &= \sqrt{(a)}\frac{\left((1+\epsilon_n) + \frac{1}{(1+\epsilon_n)}\right)}{2} \\ &= \sqrt{(a)}\left(\frac{2 + 2\epsilon_n + \epsilon_n^2}{2\,(1+\epsilon_n)}\right) \\ &= \sqrt{(a)}\left(1 + \frac{\epsilon_n^2}{2\,(1+\epsilon_n)}\right) \end{aligned}$$

Therefore,

$$\epsilon_{n+1} = \frac{\epsilon_n^2}{2\,(1+\epsilon_n)}$$

Quadratic convergence, as # correct digits doubles each step.

MULTIPLICATION ALGORITHMS

1. Naive Divide & Conquer method: $\Theta\left(d^2\right)$ time
2. Karatsuba: $\Theta\left(d^{\log_2 3}\right) = \Theta\left(d^{1.584\cdots}\right)$
3. Toom-Cook generalizes Karatsuba (break into $k \geq 2$ parts)

$$T(d) = 5T(d/3) + \Theta(d) = \Theta\left(d^{\log_3 5}\right) = \Theta\left(d^{1.465\cdots}\right)$$

4. Schönhage-Strassen - almost linear! $\Theta(d \lg d \lg \lg d)$ using FFT. All of these are in gmpy package

5. Furer (2007): $\Theta\left(n \log n 2^{O(\log^* n)}\right)$ where $\log^* n$ is iterated logarithm. # times log needs to be applied to get a number that is less than or equal to 1 .

HIGH PRECISION DIVISION

We want high precision rep of $\frac{a}{b}$

- Compute high-precision rep of $\frac{1}{b}$ first
- High-precision rep of $\frac{1}{b}$ means $\left\lfloor \frac{R}{b} \right\rfloor$ where R is large value s.t. it is easy to divide by R , Ex: $R = 2^k$ for binary representations

Newton's Method ($\frac{R}{b}$)

$$
\begin{aligned}
f(x) &= \frac{1}{x} - \frac{b}{R} \quad \left(\text{zero at } x = \frac{R}{b}\right) \\
f'(x) &= \frac{-1}{x^2} \\
\chi_{i+1} &= \chi_i - \frac{f(\chi_i)}{f'(\chi_i)} = \chi_i - \frac{\left(\frac{1}{\chi_i} - \frac{b}{R}\right)}{-1/{\chi_i}^2} \\
\chi_{i+1} &= \chi_i + {\chi_i}^2\left(\frac{1}{\chi_i} - \frac{b}{R}\right) = 2\chi_i - \frac{b{\chi_i}^2 \rightarrow \text{multiply}}{R \rightarrow \text{ easy div}}
\end{aligned}
$$

e.g.

Want $\frac{R}{b} = \frac{2^{16}}{5} = \frac{65536}{5} = 13107.2$

Try initial guess $\frac{2^{16}}{4} = 2^{14}$

$$
\begin{aligned}
\chi_0 &= 2^{14} = 16384 \\
\chi_1 &= 2 \cdot (16384) - 5(16384)^2/65536 = \underline{12}288 \\
\chi_2 &= 2 \cdot (12288) - 5(12288)^2/65536 = \underline{13056} \\
\chi_3 &= 2 \cdot (13056) - 5(13056)^2/65536 = \underline{13107}
\end{aligned}
$$

Error Analysis

$$\begin{aligned}
\chi_{i+1} &= 2\chi_i - \frac{b\chi_i^2}{R} \qquad \text{Assume } \chi_i = \frac{R}{b}(1+\epsilon_i) \\
&= 2\frac{R}{b}(1+\epsilon_i) - \frac{b}{R}\left(\frac{R}{b}\right)^2 (1+\epsilon_i)^2 \\
&= \frac{R}{b}\left((2+2\epsilon_i) - \left(1+2\epsilon_i+\epsilon_i^2\right)\right) \\
&= \frac{R}{b}\left(1-{\epsilon_i}^2\right) = \frac{R}{b}(1+\epsilon_{i+1}) \text{ where } \epsilon_{i+1} = -{\epsilon_i}^2
\end{aligned}$$

Quadratic convergence; # digits doubles at each st

Complexity of Division

One might think that the complexity of division is $\lg d$ times the complexity of multiplication given that we will have $\lg d$ multiplications in the $\lg d$ iterations required to reach precision d

However, the number of operations in division are:

$$c \cdot 1^\alpha + c \cdot 2^\alpha + c \cdot 4^\alpha + \cdots + c \cdot \left(\frac{d}{4}\right)^\alpha + c \cdot \left(\frac{d}{2}\right)^\alpha + c \cdot d^\alpha < 2c \cdot d^\alpha$$

Complexity of Computing Square Roots

If the complexity of a d -digit division is $\Theta\left(d^\alpha\right)$, then a similar summation to the one above tells us that the complexity of computing square roots is $\Theta\left(d^\alpha\right)$.

TERMINATION

Iteration: $\chi_{i+1} = \left\lfloor \frac{\chi_i + \lfloor a/\chi_i \rfloor}{2} \right\rfloor$

Do floors hurt? Does program terminate?

THINKING

本讲通过一些技巧将除法变为乘法，有一个关键是使用了 $2^k or 10^k$ 这类特定的大数，因为以它们为除数的计算十分容易。

由于Newton's Method每步的精度都使得小数点后的位数增加一倍，因此除法和开根号的计算的复杂度都和乘法相同

R12. KARATSUBA MULTIPLICATION, NEWTON'S METHOD

THINKING

这个复习课主要讨论了下边几点：

1. Karatsuba：关键的贡献是利用三个半长数的乘法解决了四个半长数的乘法（利用第三个乘法，加上或减去前两个乘法，计算了本来第三四个乘法的数值）
2. Divison：利用Newton`s Method，关键点在于将分子分母倒置，从而将困难的除法变为简单的除法
3. Initial Guess：例如对于立方根，可以先确定位数，再使用Binary Search来确定第一个数字的立方根（对于大base，小base的话直接有一个字典即可）
4. Stop Condition：$\lfloor X_i \rfloor = \lfloor X_{i+1} \rfloor$

复习课讲得更为充分，把理论的东西拆开了揉碎了讲，或者从不同的角度看讲座的内容，会有更充分的理解。（也可能是上个讲座主要看了notes，谁知道呢，接下来再看）

PSET 5

RADIX(底数) / BASE / NUMERAL SYSTEM(计数系统)

1 byte = 8 bit = 256

8-digit Binary number = 2-digit Hex number

THINKING

这个Problem Set的代码部分其实主要讨论了一个问题：对于渐近复杂度较高的算法，如果其常数系数较小，在较小输入时可能有极大的优势，因此在实践中，往往需要根据不同的输入尺度选择不同的算法。