Algorithms

Karl Schaffer, De Anza College, Math 22

Muhammad ibn Mūsā al-Khwārizmī

Origin of “algorithm.”
780-850 CE
Persian mathematician, astronomer, geographer.

Lived in Baghdad

Kitab al-Jabr wa-l-Muqabala: 1st, 2nd degree equations (origin of “Algebra.”)

Wrote about Indian decimal system

Revised Ptolemy’s *Geography.*

Horner's Method for evaluating polynomials and associated algorithms

William George Horner (1786 -1837) British mathematician

Method also known to Isaac Newton (1643-1727)

Also known to **Ch'in Chiu-Shao** (秦九韶 or 秦九劭, transcribed **Qin Jiushao** in pinyin) (ca. 1202-1261) Chinese mathematician

Ch'in Chiu-Shao

Mathematical Treatise in Nine Sections (1247):

Indeterminate analysis, military matters, surveying

Chinese remainder theorem

"Heron's formula": area of a triangle given length of three sides

Introduced zero symbol in Chinese mathematics

Techniques for solving equations, finding sums of arithmetic series, and solving linear systems

Explained how astronomical data used to construct Chinese calendar

Horner's Method for evaluating polynomials

$$p(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n,$$

Horner's Method for evaluating polynomials

$$p(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n,$$

n additions

1+2+3+...+n =
n(n+1)/2 =
$(1/2)n^2 + (1/2)n$ multiplications, if terms calculated one by one

Horner's Method for evaluating polynomials

$$p(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n,$$

$(1/2)n^2+(3/2)n$ operations

n additions
$(n^2+n)/2$ multiplications, if terms calculated one by one

Horner's Method for evaluating polynomials

$$p(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n,$$

Better: store powers of x:

3n-1 operations

n additions
n-1 multiplications for powers x^i
n multiplications for products $a_i x^i$

Horner's Method for evaluating polynomials

$$p(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n,$$

Factor:

$$= \; a_0 + x(a_1 + x(a_2 + \cdots x(a_{n-1} + b_n x) \ldots))$$

Horner's Method for evaluating polynomials

$$p(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n,$$

$$= \quad a_0 + x(a_1 + x(a_2 + \cdots x(a_{n-1} + b_n x) \ldots))$$

2n operations

n additions
n multiplications

Horner's Method for evaluating polynomials

$$= a_0 + x(a_1 + x(a_2 + \cdots x(a_{n-1} + b_n x)\ldots))$$

2n operations minimal:

Ostrowski 1954 (n additions)

Pan 1966 (n mulitplications)

-Wikipedia

Use to calculate powers of numbers efficiently:

Example:

$x^{53} = x \cdot x \cdot x \cdot ... \cdot x$ (52 multiplications)

Instead

Express 53 in binary:

$$53 = 110101_2 = 2^5 + 2^4 + 2^2 + 2^0 = 32 + 16 + 4 + 1$$

Calculate and store

$$x \cdot x = x^2$$

$$x^2 \cdot x^2 = x^4$$

$$x^4 \cdot x^4 = x^8$$

$$x^8 \cdot x^8 = x^{16}$$

$$x^{16} \cdot x^{16} = x^{32}$$

5 multiplications

$$x^{53} = x^{32} \cdot x^{16} \cdot x^4 \cdot x^1$$ 3 multiplications

Total: 5 + 3 = 8 multiplications

How do we convert 53 to binary?

Repeatedly divide 53 by 2 and store remainders:

$$53 = 2 \cdot 26 + 1$$
$$26 = 2 \cdot 13 + 0$$
$$13 = 2 \cdot 6 + 1$$
$$6 = 2 \cdot 3 + 0$$
$$3 = 2 \cdot 1 + 1$$
$$1 = 2 \cdot 0 + 1$$

Read 1s and 0s from bottom to top: 110101_2

Why is this the binary representation?

$53 = 2 \cdot 26 + 1$

$26 = 2 \cdot 13 + 0$

$13 = 2 \cdot 6 + 1$

$6 = 2 \cdot 3 + 0$

$3 = 2 \cdot 1 + 1$

$1 = 2 \cdot 0 + 1$

Read 1s and 0s from bottom to top: 110101_2

$$53 = 1+2(0+2(1+2(0+2(1+2(1)))))$$

$$= 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^5$$

Horner's Algorithm

- may be used to convert one base to another

Notice it required 6 divisions to find the binary form of 53.

How many operations to find x^{53}?

$\text{Log}_2(53)$ is between 5 and 6,
because $2^5 < 53 < 2^6$.

$\lfloor \text{Log}_2(53) \rfloor$ = "floor" of $\text{Log}_2(53)$

= greatest integer $\leq \text{Log}_2(53)$
= 5.

5+1 divisions to convert 53 to binary
5 multiplications to find $x^{32} = (x^{16})^2$ = etc.
At most 5 more multiplications to find $x^{53} = x^{32} \cdot x^{16} \cdot x^4 \cdot x^1$

Total is at most $3(5)+1 = 3\text{Log}_2(53)+1$ operations

x^n should take at most $3\text{Log}_2(n)+1$ operations

How large is $3\text{Log}_2(n)+1$?

For 100 digit number $n \approx 10^{100} \approx (2^{(10./3)})^{100}$
$\approx 2^{333}$, this takes approximately

$3 \lfloor \text{Log}_2(2^{333}) \rfloor +1 = 1000$ operations.

Who cares?
Rapid exponentiation necessary for encryption techniques, for example the RSA code.

Base conversion technique works for any base conversions, for example, convert 573 to base 8 (octal):

Repeatedly divide 573 by 8 and store remainders:

$$573 = 8 \cdot 71 + 5$$
$$71 = 8 \cdot 8 + 7$$
$$8 = 8 \cdot 1 + 0$$
$$1 = 8 \cdot 0 + 1$$

Read from bottom to top:

$$573 = 1075_8$$

$$573 = 5+8(7+8(0+8(1)))$$

$$= 5 \cdot 8^0 + 7 \cdot 8^1 + 0 \cdot 8^2 + 1 \cdot 8^3$$