Lec 1 2023 Karatsuba's Algorithm

January 4, 2023 1:41 PM

Section 4.3 of the Geddes textbook.

Assume $0 < a, b < B^n$ and $n = 2^k$ for simplicity.

Let $a = a_1 B^{n/2} + a_2$ and $b = b_1 B^{n/2} + b_2$ where $0 \le a_1, a_2, b_1, b_2 < B^{n/2}$

$B = 10$ $\quad a = \underset{a_2}{98}\,|\,\underset{a_2}{16} = 98 \cdot 10^2 + 16.$

$$a \times b = (a_1 B^{n/2} + a_2)(b_1 B^{n/2} + b_2)$$
$$= a_1 b_1 B^n + (a_1 b_2 + a_2 b_1) B^{n/2} + a_2 b_2$$

4 mults of integers of length $\frac{n}{2}$ digits and 3+ and 2 shifts.

Algorithm is to do this recursively until $n = 1$ or $a = 0$ or $b = 0$.

$a = \underset{a_1}{87}\,|\,\underset{a_2 = 0}{00}$

Example. $a = 87\,|\,65 = 87 \cdot 10^2 + 65$

$b = 65\,|\,43 = 65 \cdot 10^2 + 43$

$$a \times b = 87 \cdot 65 \cdot 10^4 + (87 \cdot 43 + 65 \cdot 65) \cdot 10^2 + 65 \cdot 13$$

$$8|7 \cdot 6|5 = 8 \cdot 6 \cdot 10^2 + \underbrace{(8 \cdot 5 + 7 \cdot 6)}_{40 + 42} \cdot 10 + 7 \cdot 5$$
$$= 4800 + 820 + 35 = \cdots$$

Let $T(n)$ be the cost of multiplying $a \times b$ of length $n = 2^k$ digits.

$$T(n) \le \underbrace{4}_{\text{4 mults}} \cdot \underbrace{T\left(\frac{n}{2}\right)}_{\text{half length}} + \underbrace{c \cdot n}_{\text{additions and shifts.}} \qquad T(1) = d.$$

$n = 2^k$

2^k $\quad T(n) \le 4\,T(n/2) + cn$

$$4\,T\left(\frac{n}{2}\right) \le 4\left(4T\left(\frac{n}{4}\right) + c\frac{n}{2}\right) = 4^2 T\left(\frac{n}{4}\right) + 2cn$$

$$4^2 T\left(\frac{n}{4}\right) \le 4^2\left(4T\left(\frac{n}{8}\right) + c\frac{n}{4}\right) = 4^3 T\left(\frac{n}{8}\right) + 4cn$$

$\vdots$

$= n$

$4^{k-1}\,T(2) \le 4^{k-1}(4T(1)+c\cdot 2) = 4^k T(1) + 2^{k-1}\cdot 2^k c$ $\quad (2^k = n)$

$4^k\,T(1) = 4^k d = (2^k)^2 \cdot d = n^2 d.$

$$T(n) = cn + 2cn + 4cn + \cdots + 2^{k-1}cn + n^2 d.$$
$$= cn(1+2+4+\cdots+2^{k-1}) + n^2 d$$
$$= cn(2^k - 1) + n^2 d$$
$$= cn^2 - cn + n^2 d$$
$$= (c+d)n^2 - cn \in O(n^2). \;!!$$

Karatsuba:

Consider $a \times b = a_1 b_1 B^n + (a_1 b_2 + a_2 b_1) B^{n/2} + a_2 b_2$

$$= a_1 b_1 B^n + \left[(a_1 - a_2)(b_2 - b_1) + a_1 b_1 + a_2 b_2\right] B^{n/2} + a_2 b_2.$$

There are 3 distinct multiplications of length $\frac{n}{2}$ plus some $+$ and $-$ and shifts which are linear in n.

$$T(n) \le 3T\left(\tfrac{n}{2}\right) + cn \quad \text{for } n > 1 \text{ and } T(1) = d.$$

Exercise: Show that $T(n) \le (2c+d)n^{\log_2 3 = 1.585} - 2cn \in O(n^{1.585})$

$$\lim_{n\to\infty} \frac{T(2n)}{T(n)} = 3$$

Example.

$a = \underset{a_1}{27}\,|\,\underset{a_2}{65}$ $\quad a \times b = 27 \cdot 43 \cdot 10^4 + \left[(27-65)(65-43) + 27 \cdot 43 + 65 \cdot 65\right] 10^2 + 65 \cdot 65.$

$b = \underset{b_1}{43}\,|\,\underset{b_2}{65}$

$a_1\ a_2\ b_1\ b_2$

$27 \cdot 43 = 2 \cdot 4 \cdot 10^2 + \left[(2-7)(3-4) + 2 \cdot 4 + 7 \cdot 3\right] \cdot 10 + 7 \cdot 3$

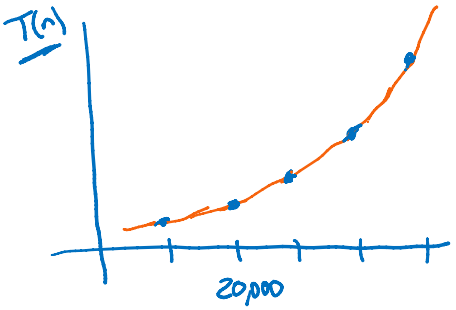
Suppose the cost of an algorithm is $T(n)$ and theoretically

Suppose the cost of an algorithm is $T(n)$ and theoretically

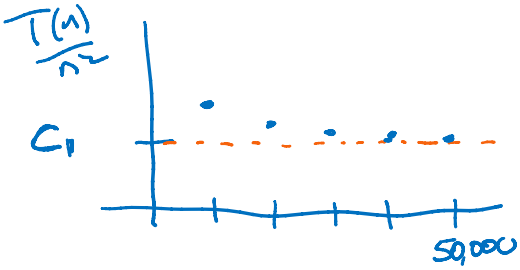
$$T(n) = c_1 n^2 + c_2 n + c_3 \in O(n^2)$$

How can we experimentally check this?

Time it for large $n =$ 10,000, 20,000, 30,000, 40,000, 50,000.



$$\lim_{n\to\infty}\left(\frac{T(n)}{n^2} = \frac{c_1 n^2 + c_2 n + c_3}{n^2} = c_1 + \frac{c_2}{n} + \frac{c_3}{n^2}\right) = c_1$$



$$T(n) = c_1 n^2 + c_2 n + c_3$$

Best way. $\lim_{n\to\infty} \frac{T(2n)}{T(n)} = \frac{c_1 4n^2 + c_2 2n + c_3}{c_1 n^2 + c_2 n + c_3} = \frac{4c_1 + 2c_2/n + c_3/n^2}{c_1 + c_2/n + c_3/n^2} = 4$

(the $2c_2/n$ and c_3/n^2 terms in the numerator, and c_2/n and c_3/n^2 in the denominator, tend to 0)

So time it for $n = 20{,}000$ and $40{,}000$ and see if $\frac{T(40{,}000)}{T(20{,}000)}$ is close to 4.