CMSC351-2023-10-27

Friday, October 27, 2023 8:07 AM

Karatsuba's Alg. For Integer Multiplication

① Goal: When multiplying two numbers, we'd like to reduce the number of single-digit multiplications (SDMs) necessary.

ex Let's multiply (37)(42). Schoolbook method:

```
  37
  42
 ---
  74
1480
----
1554
```

we did 4 SDMs. as well as some other stuff. ⊛

⊛ Could we calc this w/ fewer than 4 SDMs?

Comment: In reality in a computer this is all in binary.

② Let's Be Slick!

(A) Sps we're multiplying $(a_1a_0)(b_1b_0)$ where a_1a_0 and b_1b_0 are digits.
(So in our example $a_1=3, a_0=7, b_1=4, b_0=2$)
First we note the actual numbers are $10a_1+a_0$ and $10b_1+b_0$, so:

$$(10a_1+a_0)(10b_1+b_0) = 100a_1b_1 + 10(a_0b_1+a_1b_0) + a_0b_0 \quad ⊛$$

4 SDMS HERE!

let's reduce this to 3.
observe that $(a_1+a_0)(b_1+b_0) = a_1b_1 + a_0b_1 + a_1b_0 + a_0b_0$

so $a_0b_1 + a_1b_0 = (a_1+a_0)(b_1+b_0) - a_1b_1 - a_0b_0$

So then plug this into ⊛ to get

$$(10a_1+a_0)(10b_1+b_0) = 100a_1b_1 + 10[(a_1+a_0)(b_1+b_0) - a_1b_1 - a_0b_0] + a_0b_0$$

we now have 2 SDMs a_1b_1 and a_0b_0
and 1 mult $(a_1+a_0)(b_1+b_0)$ which may be a SDM but

we now have 2 SDMs $a_1 a_0$ and $a_0 b_0$
and 1 mult $(a_1+a_0)(b_1+b_0)$ which may be a SDM but in any case it's easier than a standard 2-digit mult.
for now treat $(a_1+a_0)(b_1+b_0)$ as a SDM.

This to do $(a_1 a_0)(b_1 b_0)$ we need, in total:

- 3 SDM (ish)
- 6 +/- w/ up to 4 digits each
- mult by 100 and by 10. Thats 3 decimal shifts total

(b) Sps were mult two 4-digit numbers.
Represent as $(A_1 A_0)(B_1 B_0)$ w/ A_1, A_0, B_1, B_0 each 2-digits.
(for ex. (4386)(9541) we'd have $A_1=43, A_0=86, B_1=95, B_0=41$)
the same process as above yields:

$$(10^2 A_1 + A_0)(10^2 B_1 + B_0) = 10000 A_1 B_1 + 100[(A_1+A_0)(B_1+B_0) - A_1 B_1 - A_0 B_0] + A_0 B_0$$

here we have:

- 3 double-digit mult (ish)
- 6 +/- w/ up to 8 digits each
- mult by 10000 and by 100. Thats 6 decimal shifts total.

(c) Now generalize: Sps $A_1 A_0$ and $B_1 B_0$ are each n-digit #s w/ n=even.
the same process yields

$$(10^{n/2} A_1 + A_0)(10^{n/2} B_1 + B_0) = 10^n A_1 B_1 + 10^{n/2}[(A_1+A_0)(B_1+B_0) - A_1 B_1 - A_0 B_0] + A_0 B_0$$

here we have

- 3 mult. w/ $\frac{n}{2}$ digits (ish)
- 6 +/- w/ up to $2n$ digits each (*)
- $n+\frac{n}{2}$ decimal shifts total. (**)

(*) This process is $\Theta(2n)$ each, 6 times. so $\Theta(n)$
(**) Each shift is $\Theta(1)$ so total is $\Theta(n)$
Together (*) and (**) are $\Theta(n)$.

③ Creating an Algorithm: Informally given a product AB, each n digits
[illegible] split each in half and apply the above method

③ Creating an Algorithm: Informally given a product AB, each n digits we'll split each in half and apply the above method.
For each of our 3 new mult. do the same (recurse!)
If $T(n)$ = time required to do this w/ two n-digit numbers, then:

$$T(n) = \underbrace{3T\left(\frac{n}{2}\right)}_{\text{3 mult of integers w/ half as many digits!}} + \underbrace{\Theta(n)}_{\text{6 +/- and decimal shifts!}}$$

By master thm: $\log_b a = \log_2 3$ and $c = 1$ b/c of $\Theta(n)$
$\log_2 3 = \lg 3 > 1$ so Case 1!

Thus $T(n) = \Theta(n^{\log_2 3}) = \Theta(n^{\lg 3}) \approx \Theta(n^{1.5849625})$

Compare: w/ schoolbook method, two n-digit numbers requires n^2 sdm (and some +) which ends up being $\Theta(n^2)$

Aha! If we do this we get better time complexity!

Next: Write pseudocode and iron out some details!