Multiply Integers

$$1234 \times 5678$$

Quiz
What's a basic operation in the classic pen-paper procedure?

- ➪ **Classic procedure**
- ➪ **Basic operation**
- ➪ **Efficiency (brute force)**

Multiply Integers
Another Procedure

Quiz
Why would x, x^2 not be considered in multiplication count?

Try 2-digit integers (n=2)

$23 = 2 \times 10^1 + 3$

$$(ax + b)(cx + d) = acx^2 + (\cancel{ad} + \cancel{bc})x + bd$$

C_1 (= ac), C_2 (= bd)

$$(a + b) \times (c + d) - ac - bd \quad \} \; C_3$$

Each constant involves 1 novel multiplication.

$$C_1 \times 10^n + C_3 \times 10^{n/2} + C_2$$

Exercise
Determine the 3 constants.
Repeat for 12×34.

Example

a, b: $(2 \times 10^1 + 3)$

$$23 \times 14 = 322$$

c, d: $(1 \times 10^1 + 4)$

❶ $(2 \times 1) \times 10^2 +$

$[(2 + 3) \times (1 + 4)$ ❸ $- (2 \times 1) - (3 \times 4)] \times 10^1 +$

(3×4) ❷

Multiply Integers
Bigger Example

Exercise
Determine *C*1, *C*2, *C*3 and verify the formula (sub figure) indeed gives the correct answer.

Apply to 4-digit integers

$$1234 \times 5678 = 7006652$$

$$(12 \times 10^2 + 34) \times (56 \times 10^2 + 78) =$$

$$C_1 \times 10^n + C_3 \times 10^{n/2} + C_2$$

$$\underset{❶}{(12 \times 56)} \times 10^4 +$$

$$\left[(12 + 34) \underset{❸}{\times} (56 + 78) - (❶) - (❷)\right] \times 10^2 +$$

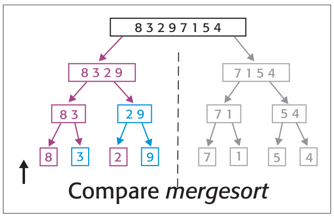
$$\underset{❷}{(34 \times 78)}$$

Repeat previous 2-digit multiplication

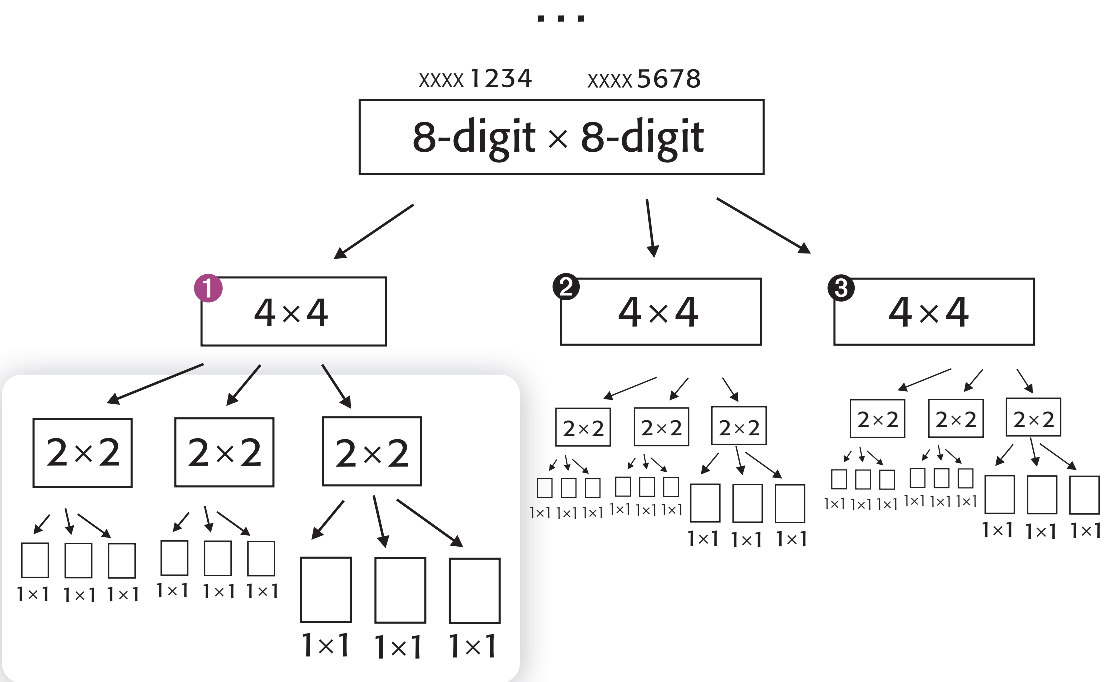
cs223fig25.cdr
Monday, April 4, 2022 10:25:49 AM
Color profile: Disabled

Composite Default screen

Multiply Integers
Divide-Conquer



Instance replaced by 3 smaller ones every time it's reduced by half (a constant factor of 2).



Quiz
How many 1-digit mults are performed for n=8 (count them)? Compare to classic.

Exercise
Trace the example down to 1-digit mults (replace x's by 0's). Verify result 7006652.

Divide-Conquer Multipy Performance

$C_1 \times 10^n + C_3 \times 10^{n/2} + C_2$

Note how 3 const combine (no basic ops needed).

Quiz
What if integer length n is not a power of 2?

➪ Write recurrence

Exercise
Verify that mathematical result matches count from previous slide for n=8?

➪ Backward substitutions

Size reduction factor ***b*** need not match number of instances ***a*** into which problem is divided.

➪ Check Master Theorem

$$T(n) \in \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{if } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

?

Quiz
What if addition is chosen as basic operation?

➪ Choice of basic operation

Divide-Conquer Multipy
Conclusions

➪ Asymptotic efficiency

$O(n^{1.585})$

➪ **Pioneering, Karatsuba 1960**

$O(n^{1.465})$

$O(n \log n \log \log n)$

Later optimizations lead to unconfirmed suspicion of optimal lower limit of *n*log *n*.

???

$\Omega(n \log n)$

➪ **Improvements**

➪ **Asymptotic advantage**

➪ **Applications**

Quiz
What's the main application of divide-conquer multiplication of large integers? (Hint: check textbook).

Divide-Conquer Multiplication Strategy

Find a shortcut for performing a basic case, then use divide-conquer to apply repeatedly and hope for enough savings to add up

(Unlike brute force where we make no such effort)

Multiply Matrices

Quiz
How many element multiplications are performed in this case?

Basic case

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \times \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} a_{00}b_{00} + a_{01}b_{10} & a_{00}b_{01} + a_{01}b_{11} \\ a_{10}b_{00} + a_{11}b_{10} & a_{10}b_{01} + a_{11}b_{11} \end{bmatrix}$$

Shortcut

Product of 2×2 matrices can be computed via <u>7 multiplications</u> instead of the 8 by definition-based n^3 method

Divide-Conquer Multiplication
Strassen's Algorithm

Do the exercise

- **Basic 2×2 case (m_1 .. m_7)**
- **Repeatedly (like Karatsuba)**
  - Divide $n \times n$ into $^n/_2 \times {}^n/_2$ submatrices
  - Treat submatrices as numbers
  - Compute 7 product submatrices
- **Performance and results** ??? $\Omega(n^2)$

Similar to diagram in Slide 4, each divide by 2 step results in 7 product matrices.

Quiz
Determine constants *a*,*b* in Master Theorem?

Quiz
How many additions are needed compared to brute-force algorithm?