5.5 Integer Multiplication

Integer Arithmetic

Add. Given two n-digit integers a and b, compute a + b.

- O(n) bit operations.

Multiply. Given two n-digit integers a and b, compute a × b.

Q. How many bit operations does a brute-force multiply need?

A.

1	1	1	1	1	1	0	1	
	1	1	0	1	0	1	0	1
+	0	1	1	1	1	1	0	1
1	0	1	0	1	0	0	1	0

Add

TU Delft

Integer Arithmetic

Add. Given two n-digit integers a and b, compute a + b.

- O(n) bit operations.

Multiply. Given two n-digit integers a and b, compute a × b.

Q. How many bit ops. does a brute-force multiply need?

A. Brute force solution: $\Theta(n^2)$ bit operations.

1	1	1	1	1	1	0	1	
	1	1	0	1	0	1	0	1
+	0	1	1	1	1	1	0	1
1	0	1	0	1	0	0	1	0

Add

					1	1	0	1	0	1	0	1			
				*	0	1	1	1	1	1	0	1			
					1	1	0	1	0	1	0	1			
				0	0	0	0	0	0	0	0				
			1	1	0	1	0	1	0	1					
		1	1	0	1	0	1	0	1						
	1	1	0	1	0	1	0	1							
	1	1	0	1	0	1	0	1							
0	0	0	0	0	0	0	0								
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1

Multiply

Divide-and-Conquer Multiplication: Warmup

Q. How to use divide and conquer to multiply two n-digit integers? (1 min)

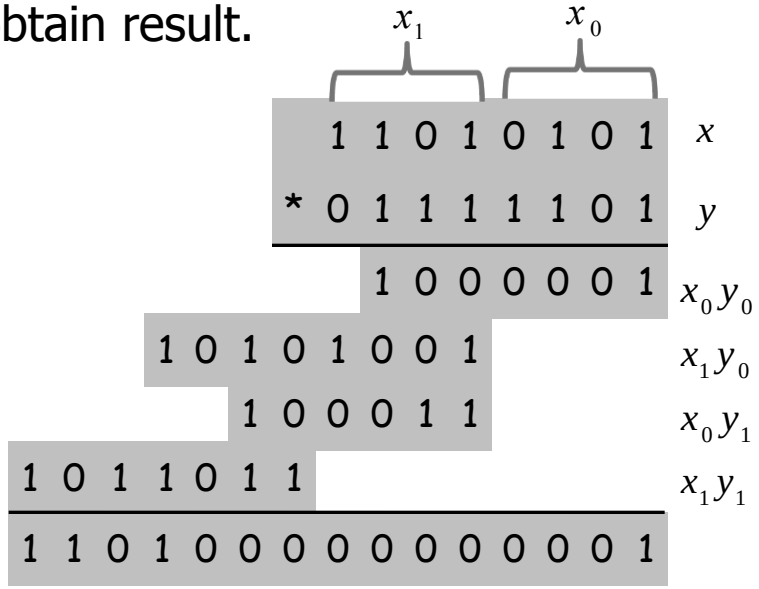
TU Delft

Divide-and-Conquer Multiplication: Warmup

To multiply two n-digit integers:

- Multiply four ½n-digit integers.
- Add two ½n-digit integers, and shift to obtain result.

$$
\begin{aligned}
x &= 2^{n/2} \cdot x_1 + x_0 \\
y &= 2^{n/2} \cdot y_1 + y_0 \\
xy &= \left(2^{n/2} \cdot x_1 + x_0\right)\left(2^{n/2} \cdot y_1 + y_0\right) \\
&= 2^n \cdot x_1 y_1 + 2^{n/2} \cdot \left(x_1 y_0 + x_0 y_1\right) + x_0 y_0
\end{aligned}
$$



$$
\mathrm{T}(n) = \underbrace{4\,T(n/2)}_{\text{recursive calls}} + \underbrace{\Theta(n)}_{\text{add, shift}}
$$

↑ assumes n is a power of 2

Master method:

$$a = 4,\ b = 2,\ n^{\log_b a} = n^2$$

$$f(n) = O(n^{2-c})$$

$$\mathrm{T}(n) = \Theta(n^2)$$

Karatsuba Multiplication

To multiply two n-digit integers:

- Add two ½n digit integers.
- Multiply three ½n-digit integers.
- Add, subtract, and shift ½n-digit integers to obtain result.

$$
\begin{aligned}
x &= 2^{n/2} \cdot x_1 + x_0 \\
y &= 2^{n/2} \cdot y_1 + y_0 \\
xy &= 2^{n} \cdot x_1 y_1 + 2^{n/2} \cdot (x_1 y_0 + x_0 y_1) + x_0 y_0 \\
&= 2^{n} \cdot x_1 y_1 + 2^{n/2} \cdot ((x_1 + x_0)(y_1 + y_0) - x_1 y_1 - x_0 y_0) + x_0 y_0 \\
&\qquad\quad A \qquad\qquad\qquad B \qquad\qquad A \quad C \qquad C
\end{aligned}
$$

Theorem. [Karatsuba-Ofman, 1962] Can multiply two n-digit integers in $O(n^{1.585})$ bit operations.

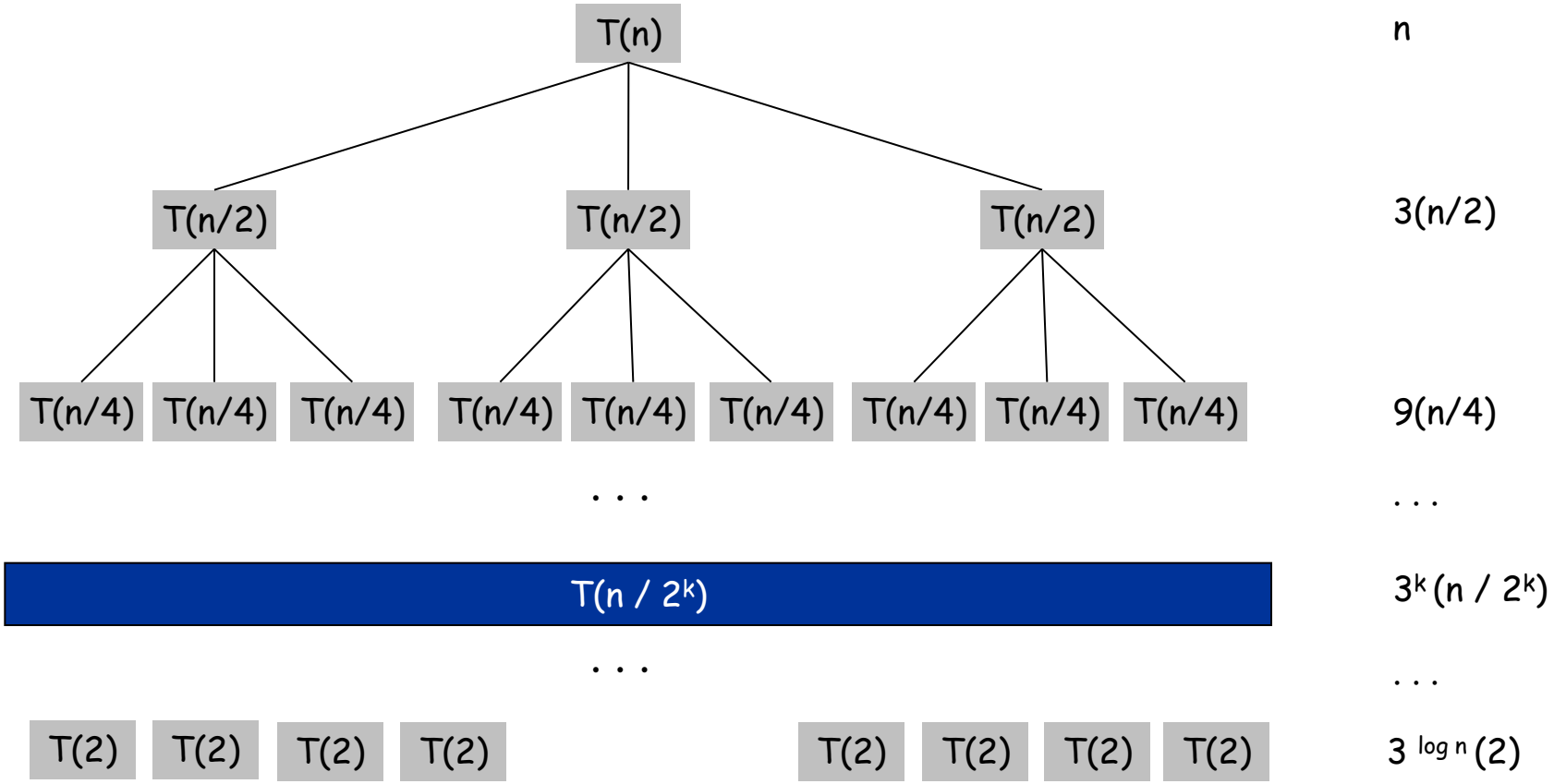
$$
T(n) \le \underbrace{T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + T(1 + \lceil n/2 \rceil)}_{\text{recursive calls}} + \underbrace{\Theta(n)}_{\text{add, subtract, shift}}
$$

$$
\Rightarrow T(n) = O(n^{\log_2 3}) = O(n^{1.585})
$$

Karatsuba: Recursion Tree

$$T(n) = \begin{cases} 0 & \text{if } n = 1 \\ 3T(n/2) + n & \text{otherwise} \end{cases}$$

$$T(n) = \sum_{k=0}^{\log_2 n} n \left(\tfrac{3}{2}\right)^k = \frac{\left(\frac{3}{2}\right)^{1+\log_2 n} - 1}{\frac{3}{2} - 1} = 3n^{\log_2 3} - 2$$



Master Method Summarized

Given a recurrence of the form $T(n) = aT(n/b) + f(n)$

We can distinguish three common cases:

1. Running time dominated by cost at leaves:
   if $f(n) = O\left(n^{\log_b a - \varepsilon}\right)$ then $T(n) = \Theta\left(n^{\log_b a}\right)$

2. Running time evenly distributed throughout the tree:
   if $f(n) = \Theta\left(n^{\log_b a}\right)$ then $T(n) = \Theta\left(n^{\log_b a} \log n\right)$

3. Running time dominated by cost at root:
   if $f(n) = \Omega\left(n^{\log_b a + \varepsilon}\right)$ then $T(n) = \Theta\left(f(n)\right)$

If f(n) satisfies regularity condition: $a\, f(n/b) \leq c\, f(n)$ for some $c < 1$ (polynomials always do)

The master method cannot solve every recurrence of this form.

TU Delft

Examples

Q. Use the master method to solve the following recurrence relation:

$$T(n) = 3T(n/2) + n$$

A. Analysis:

$$a = 3,\ b = 2; \qquad n^{\log_2 3} = n^{1.585}$$

$$f(n) = \Theta(n)$$

This is case 1, $f(n) = O\left(n^{\log_b a - c}\right)$, so:

$$T(n) = \Theta\left(n^{\log_b a}\right) = \Theta\left(n^{\log_2 3}\right)$$

TU Delft

Examples

Q. Use the master method to solve the following recurrence relation:

$$T(n) = 3T(n/4) + n \log n$$

WARNING:
Check regularity condition if case 3.

A. Analysis:

$$a = 3,\ b = 4; \qquad n^{\log_4 3} = n^{0.793}$$

$$f(n) = \Theta(n \log n)$$

TU Delft

Examples

Q. Use the master method to solve the following recurrence relation:

$$T(n) = 3T(n/4) + n \log n$$

WARNING:
Check regularity condition if case 3.

A. Analysis:

$$a = 3,\ b = 4; \qquad n^{\log_4 3} = n^{0.793}$$

$$f(n) = \Theta(n \log n)$$

This is case 3, $f(n) = \Omega\left(n^{\log_b a + c}\right)$, so:

$$T(n) = \Theta\left(f(n)\right) = \Theta\left(n \log n\right)$$

Check regularity condition:

$$af(n/b) = 3(n/4)\log(n/4) \le (3/4)n \log n = cf(n)$$

OK, for example for c=3/4 (and this c < 1)