CS11313 - **Spring** 2023

Design & Analysis *of* Algorithms

Integer Multiplication

more Divide & Conquer Examples!

Ibrahim Albluwi

Overview

Problem. Given two positive integers x and y of length n digits each, find $x \times y$.

Rules of the game.

- Adding or subtracting two n-digit numbers requires $\Theta(n)$ operations.
- Single-digit multiplication requires constant time.
  (can be precomputed)
- Multiplying a decimal number by 10 requires constant time.
  (shift-left)

Goal. See an example of a divide and conquer algorithm that is not related to searching or sorting.

Attempt # 1: Repeated Addition

Problem. Given two positive integers x and y of length n digits each, find $x \times y$.

Solution 1. Repeated addition.

Example: `200 x 300 = 300 + 300 + ...` (`200` times)

Quiz 1

Problem. Given two positive integers x and y of length n decimal digits each, how many single digit operations are performed if $x \times y$ is computed using repeated addition?

A. $\Theta(n)$

B. $\Theta(n \log n)$

C. $\Theta(n^2)$

D. $\Theta(2^n)$

E. None of the above

Quiz 1

Problem. Given two positive integers x and y of length n decimal digits each, how many single digit operations are performed if $x \times y$ is computed using repeated addition?

A. $\Theta(n)$

B. $\Theta(n \log n)$

C. $\Theta(n^2)$

D. $\Theta(2^n)$

E. None of the above

- The largest decimal number representable using n digits is $10^n - 1$
- Therefore, y is added $\sim 10^n$ times in the worst case.
- Each addition involves $\Omega(n)$ single digit operations.
- **Total** $= \Omega(n10^n)$ single digit operations.

Attempt # 2: Long Multiplication

Problem. Given two positive integers x and y of length n digits each, find $x \times y$.

Solution 1. Repeated addition.

Example: `200 x 300 = 300 + 300 + ...` (200 times)

$\sim 10^n$ additions each involving $\Omega(n)$ digits = $\Omega(n10^n)$ single digit operations.

The largest decimal number representable using n digits is $10^n - 1$

Solution 2. Long Multiplication.

Example: `2311 x 4301 =`

$$
\begin{array}{rcccc}
 & 2 & 3 & 1 & 1 \\
\times & 4 & 3 & 0 & \mathbf{1} \\
\hline
 & 2 & 3 & 1 & 1
\end{array}
$$

Attempt # 2: Long Multiplication

Problem. Given two positive integers x and y of length n digits each, find $x \times y$.

Solution 1. Repeated addition.

Example: `200 x 300 = 300 + 300 + ...` (`200` times)

$\sim 10^n$ additions each involving $\Omega(n)$ digits = $\Omega(n10^n)$ single digit operations.

The largest decimal number representable using n digits is $10^n - 1$

Solution 2. Long Multiplication.

Example: `2311 x 4301 =`

$$
\begin{array}{rcccc}
 & 2 & 3 & 1 & 1 \\
\times & 4 & 3 & \mathbf{0} & 1 \\
\hline
 & 2 & 3 & 1 & 1 \\
0 & 0 & 0 & 0 & \\
\end{array}
$$

Attempt # 2: Long Multiplication

Problem. Given two positive integers x and y of length n digits each, find $x \times y$.

Solution 1. Repeated addition.

Example: `200 x 300 = 300 + 300 + ...` (`200` times)

$\sim 10^n$ additions each involving $\Omega(n)$ digits = $\Omega(n10^n)$ single digit operations.

The largest decimal number representable using n digits is $10^n - 1$

Solution 2. Long Multiplication.

Example: `2311 x 4301 =`

```
        2 3 1 1

x       4 3 0 1
---------------
        2 3 1 1
      0 0 0 0
    6 9 3 3
```

Attempt # 2: Long Multiplication

Problem. Given two positive integers x and y of length n digits each, find $x \times y$.

Solution 1. Repeated addition.

Example: `200 x 300 = 300 + 300 + …` (200 times)

$\sim 10^n$ additions each involving $\Omega(n)$ digits = $\Omega(n10^n)$ single digit operations.

The largest decimal number representable using n digits is $10^n - 1$

Solution 2. Long Multiplication.

Example: `2311 x 4301 =`

$\Theta(n^2)$ single digit operations.

Can we do better?

```
          2 3 1 1
        x 4 3 0 1
        ---------
          2 3 1 1
        0 0 0 0
      6 9 3 3
  + 9 2 4 4
  ---------------
    9 9 3 9 6 1 1
```

Attempt # 4: A Divide & Conquer Algorithm

Problem. Given two positive integers x and y of length n digits each, find $x \times y$.

Observation 1. A positive integer x of length n digits can be split into 2 halves as follows:

$$\underbrace{4\ 5\ 3}_{a}\ \underbrace{9\ 6\ 1}_{b}$$

Left half: Shift right $\frac{n}{2}$ steps: 0 0 0 4 5 3

Right half: Shift left $\frac{n}{2}$ steps: 9 6 1 0 0 0

then Shift right $\frac{n}{2}$ steps 0 0 0 9 6 1

> The number x can be written as $a \bullet 10^{\frac{n}{2}} + b$

Attempt # 4: A Divide & Conquer Algorithm

Problem. Given two positive integers x and y of length n digits each, find $x \times y$.

Observation 1. A positive integer x of length n digits can be split into 2 halves as follows:

$$\underbrace{4\ 5\ 3}_{a}\ \underbrace{9\ 6\ 1}_{b}$$

Left half: Shift right $\frac{n}{2}$ steps: `0 0 0 4 5 3`

Right half: Shift left $\frac{n}{2}$ steps: `9 6 1 0 0 0`
then Shift right $\frac{n}{2}$ steps `0 0 0 9 6 1`

> The number x can be written as $a \bullet 10^{\frac{n}{2}} + b$

Observation 2. $x \times y$ can be written as: $(a \bullet 10^{\frac{n}{2}} + b) \times (c \bullet 10^{\frac{n}{2}} + d)$

left half of y

right half of y

Attempt # 4: A Divide & Conquer Algorithm

Problem. Given two positive integers x and y of length n digits each, find $x \times y$.

Observation 1. A positive integer x of length n digits can be split into 2 halves as follows:

$$\underbrace{4\ 5\ 3}_{a}\ \underbrace{9\ 6\ 1}_{b}$$

Left half: Shift right $\frac{n}{2}$ steps: 0 0 0 4 5 3

Right half: Shift left $\frac{n}{2}$ steps: 9 6 1 0 0 0

then Shift right $\frac{n}{2}$ steps 0 0 0 9 6 1

> The number x can be written as $a \bullet 10^{\frac{n}{2}} + b$

Observation 2. $x \times y$ can be written as: $(a \bullet 10^{\frac{n}{2}} + b) \times (c \bullet 10^{\frac{n}{2}} + d)$

$$= ac \bullet 10^n + ad \bullet 10^{\frac{n}{2}} + bc \bullet 10^{\frac{n}{2}} + bd$$

$$= \underline{ac} \bullet 10^n + (\underline{ad} + \underline{bc}) \bullet 10^{\frac{n}{2}} + \underline{bd}$$

> Multiplying two n-digit numbers requires **4** multiplications between numbers of length $\frac{n}{2}$ each!

Attempt # 4: A Divide & Conquer Algorithm

$$= \underline{ac} \bullet 10^{n} + (\underline{ad} + \underline{bc}) \bullet 10^{\frac{n}{2}} + \underline{bd}$$

Multiplying two n-digit numbers requires **4** multiplications between numbers of length $\frac{n}{2}$ each!

Running Time. $T(n) = 4T(\frac{n}{2}) + cn$

to shift and perform 3
addition operations

Attempt # 4: A Divide & Conquer Algorithm

$$= \underline{ac} \bullet 10^n + (\underline{ad} + \underline{bc}) \bullet 10^{\frac{n}{2}} + \underline{bd}$$

Multiplying two n-digit numbers requires **4** multiplications between numbers of length $\frac{n}{2}$ each!

Running Time. $T(n) = 4T(\frac{n}{2}) + cn \quad = \quad \Theta(n^2)$ 😁 using the master method

🤔 Can we reduce the number of subproblems?

Attempt # 4: A Divide & Conquer Algorithm

$$= \underline{ac} \bullet 10^{n} \;+\; (\underline{ad} + \underline{bc}) \bullet 10^{\frac{n}{2}} \;+\; \underline{bd}$$

Multiplying two n-digit numbers requires **4** multiplications between numbers of length $\frac{n}{2}$ each!

Running Time. $T(n) = 4T(\frac{n}{2}) + cn \quad = \quad \Theta(n^2)$ using the master method

Can we reduce the number of subproblems?

Karatsuba

ENTER

Karatsuba

Attempt # 4: A Divide & Conquer Algorithm

$$= \underline{ac} \bullet 10^{n} \; + \; (\underline{ad} + \underline{bc}) \bullet 10^{\frac{n}{2}} \; + \; \underline{bd}$$

Multiplying two n-digit numbers requires **4** multiplications between numbers of length $\frac{n}{2}$ each!

Running Time. $T(n) = 4T(\frac{n}{2}) + cn \quad = \quad \Theta(n^2)$

Can we reduce the number of subproblems?

Karatsuba

Observation. If we know what $(a+b)(c+d)$ we what $(ad+bc)$ is!

Explanation. $(a+b)(c+d) \;=\; ac + \underbrace{ad + bc}_{\text{we want this sum}} + bd$

we know these

Attempt # 4: A Divide & Conquer Algorithm

$$= \underline{ac} \bullet 10^n \; + \; (\underline{ad} + \underline{bc}) \bullet 10^{\frac{n}{2}} \; + \; \underline{bd}$$

Multiplying two n-digit numbers requires **4** multiplications between numbers of length $\frac{n}{2}$ each!

Running Time. $T(n) = 4T(\frac{n}{2}) + cn \quad = \quad \Theta(n^2)$

Can we reduce the number of subproblems?

Karatsuba

Observation. If we know what $(a+b)(c+d)$ we what $(ad+bc)$ is!

Explanation. $(a+b)(c+d) \;\; = \;\; ac + ad + bc + bd$

$$(a+b)(c+d) - (ac+bd) = (ac + ad + bc + bd) - (ac + bd)$$

Attempt # 4: A Divide & Conquer Algorithm

$$= \underline{ac} \bullet 10^n \;+\; (\underline{ad} + \underline{bc}) \bullet 10^{\frac{n}{2}} \;+\; \underline{bd}$$

Multiplying two n-digit numbers requires **4** multiplications between numbers of length $\frac{n}{2}$ each!

Running Time. $T(n) = 4T(\frac{n}{2}) + cn \quad = \quad \Theta(n^2)$

Can we reduce the number of subproblems?

Karatsuba

Observation. If we know what $(a+b)(c+d)$ we what $(ad+bc)$ is!

Explanation. $(a+b)(c+d) \;=\; ac + ad + bc + bd$

bingo!

$$(a+b)(c+d) - (ac+bd) = (\cancel{ac} + ad + bc + \cancel{bd}) - (\cancel{ac} + \cancel{bd}) \boxed{= (ad+bc)}$$

only **one** multiplication is needed to find $(ad+bc)$

Attempt # 4: A Divide & Conquer Algorithm

$$= \underline{ac} \bullet 10^n \ + \ (\underline{ad} + \underline{bc}) \bullet 10^{\frac{n}{2}} \ + \ \underline{bd}$$

Multiplying two n-digit numbers requires **4** multiplications between numbers of length $\frac{n}{2}$ each!

Running Time. $T(n) = 4T(\frac{n}{2}) + cn \quad = \quad \Theta(n^2)$

Can we reduce the number of subproblems?

Karatsuba

Observation. If we know what $(a+b)(c+d)$ we what $(ad+bc)$ is!

Explanation. $(a+b)(c+d) \ = \ ac + ad + bc + bd$

bingo!

$$(a+b)(c+d) - (ac+bd) = (\cancel{ac} + ad + bc + \cancel{bd}) - (\cancel{ac} + \cancel{bd}) \boxed{= (ad+bc)}$$

only **one** multiplication is needed to find $(ad+bc)$

$$x \times y \ = ac \bullet 10^n \ + \ \boxed{(ad+bc)} \bullet 10^{\frac{n}{2}} \ + \ bd$$

$$= ac \bullet 10^n \ + \ \boxed{((a+b)(c+d) - (ac+bd))} \bullet 10^{\frac{n}{2}} \ + \ bd$$

Attempt # 4: A Divide & Conquer Algorithm

$$= \underline{ac} \bullet 10^n + (\underline{ad} + \underline{bc}) \bullet 10^{\frac{n}{2}} + \underline{bd}$$

Multiplying two n-digit numbers requires **4** multiplications between numbers of length $\frac{n}{2}$ each!

Running Time. $T(n) = 4T(\frac{n}{2}) + cn \quad = \quad \Theta(n^2)$

Can we reduce the number of subproblems?

Karatsuba

Observation. If we know what $(a+b)(c+d)$ we what $(ad+bc)$ is!

bingo!

Explanation. $(a+b)(c+d) \;=\; ac + ad + bc + bd$

$$(a+b)(c+d) - (ac+bd) = (\cancel{ac} + ad + bc + \cancel{bd}) - (\cancel{ac} + \cancel{bd}) \boxed{= (ad+bc)}$$

only **one** multiplication is needed to find $(ad+bc)$

$$x \times y = ac \bullet 10^n + (ad+bc) \bullet 10^{\frac{n}{2}} + bd$$

$$= ac \bullet 10^n + (((a+b)(c+d)) - (ac+bd)) \bullet 10^{\frac{n}{2}} + bd$$

only **three** multiplications are needed to find $x \times y$

Karatsuba's Algorithm: Implementation

```
// assuming x and y have n digits each and n is a power of 2
MULTIPLY(x, y):
   n = number of digits in x and y
   if (n == 1) return x * y

   a = n/2 left-most  digits of x
   b = n/2 right-most digits of x
   c = n/2 left-most  digits of y
   d = n/2 right-most digits of y

1  ac   = MULTIPLY(a,   c)
2  bd   = MULTIPLY(b,   d)
3  temp = MULTIPLY(a+b, c+d)

  return [ac * 10^n] + [(temp - ac - bd) * 10^(n/2)] + bd
          1             3      1    2                  2
```

Karatsuba's Algorithm: Implementation and Analysis

```
// assuming x and y have n digits each and n is a power of 2
MULTIPLY(x, y):
   n = number of digits in x and y
   if (n == 1) return x * y

   a = n/2 left-most  digits of x
   b = n/2 right-most digits of x
   c = n/2 left-most  digits of y
   d = n/2 right-most digits of y

   ac   = MULTIPLY(a,   c)
   bd   = MULTIPLY(b,   d)
   temp = MULTIPLY(a+b, c+d)

  return [ac * 10^n] + [(temp - ac - bd) * 10^(n/2)] + bd
```

Running Time.

$T(n) = 3T(\frac{n}{2}) + cn$ —— time to add, subtract, shift, etc.

Karatsuba's Algorithm: Implementation and Analysis

```
// assuming x and y have n digits each and n is a power of 2
MULTIPLY(x, y):
   n = number of digits in x and y
   if (n == 1) return x * y

   a = n/2 left-most  digits of x
   b = n/2 right-most digits of x
   c = n/2 left-most  digits of y
   d = n/2 right-most digits of y

   ac   = MULTIPLY(a,   c)
   bd   = MULTIPLY(b,   d)
   temp = MULTIPLY(a+b, c+d)

  return [ac * 10^n] + [(temp - ac - bd) * 10^(n/2)] + bd
```

Running Time.

$$T(n) = 3T(\tfrac{n}{2}) + cn \;= \Theta(n^{\log_2 3})$$
$$= O(n^{1.59})$$

using the master method

Karatsuba's Algorithm: Implementation and Analysis

```
// assuming x and y have n digits each and n is a power of 2
MULTIPLY(x, y):
   n = number of digits in x and y
   if (n == 1) return x * y

   a = n/2 left-most  digits of x
   b = n/2 right-most digits of x
   c = n/2 left-most  digits of y
   d = n/2 right-most digits of y

   ac   = MULTIPLY(a,   c)
   bd   = MULTIPLY(b,   d)
   temp = MULTIPLY(a+b, c+d)

  return [ac * 10^n] + [(temp - ac - bd) * 10^(n/2)] + bd
```

Running Time.

$$T(n) = 3T(\tfrac{n}{2}) + cn \;= \Theta(n^{\log_2 3})$$
$$= O(n^{1.59})$$

Is it worth the effort?

Yes if n is large (>100)

No if n is small.

Integer Multiplication: A Race for Efficiency!

year	algorithm	bit operations
12xx	**grade school**	$O(n^2)$
1962	**Karatsuba–Ofman**	$O(n^{1.585})$
1963	**Toom-3, Toom-4**	$O(n^{1.465}),\ O(n^{1.404})$
1966	**Toom–Cook**	$O(n^{1+\varepsilon})$
1971	**Schönhage–Strassen**	$O(n \log n \cdot \log \log n)$
2007	**Fürer**	$n \log n \, 2^{O(\log^* n)}$
2018	**Harvey–van der Hoeven**	$O(n \log n \cdot 2^{2 \lg^* n})$
	???	$O(n)$

Remark. GNU Multiple Precision library uses one of first five algorithms depending on n.

GMP
«Arithmetic without limitations»

slide by Kevin Wayne