Lec. 19: Integer Multiplication Date: 16/01/23

- We need to multiply 2 n-digit (or n-bit) numbers.
- Answer can be 2n digits long.
- Traditional algorithm is efficient ($O(n^2)$).

$n^2$ multiplications plus upto $n^2+n-2$ additions.

So $O(n^2)$ effort.

```
        ← n
      999
    × 999
    -----
     8991      ↑
    8991×      | n
   8991××      ↓
   ------
   998001   ← Length = 6 = 2n
```

$0+1+2+2+1+0$ Normal additions $+$ upto $2n-2$ carry-over additions

$= 2(n-1+\dots+1+0) + 2n-2$
$= (n-1)n + 2n - 2 = n^2+n-2$ additions at max.

- A recursive algorithm.

$x = x_1 \times 10^{n/2} + x_0$ ← Note: 10 can be replaced by the base of the number system you are working in.

$y = y_1 \times 10^{n/2} + y_0$

$$\underbrace{x\times y}_{2n} = \underbrace{\underbrace{\overset{A}{\underbrace{x_1}_{n/2}\underbrace{y_1}_{n/2}}}_{n}\times \underbrace{10^n}_{n}}_{2n} + \underbrace{\underbrace{\underbrace{(\overset{B}{\underbrace{x_1}_{n/2}\underbrace{y_0}_{n/2}} + \overset{C}{\underbrace{x_0}_{n/2}\underbrace{y_1}_{n/2}})}_{n+1}\times \underbrace{10^{n/2}}_{n/2}}_{n+\frac{n}{2}+1} + \underbrace{\overset{D}{\underbrace{x_0}_{n/2}\underbrace{y_0}_{n/2}}}_{n}}_{n+1}$$

RecursiveMultiply(x, y, b) ← base

    n = # digits in x
    if n < 2
        return x × y
    else
        x_1 = first n/2 digits of x
        x_0 = last n/2 digits of x
        y_1 = first n/2 digits of y
        y_0 = last n/2 digits of y
        A = RecursiveMultiply(x_1, y_1, b)
        B = RecursiveMultiply(x_1, y_0, b)
        C = RecursiveMultiply(x_0, y_1, b)
        D = RecursiveMultiply(x_0, y_0, b)
    return A × b^n + (B+C) × b^{n/2} + D

4 mults of $n/2$ digit numbers

$T(n) \leq 4\,T(n/2) + cn$

$T(n) = O(n^{\log_2 4}) = O(n^2)$

4 multiplications of $\frac{n}{2}$ digit numbers

$T(n) \leq 4T(\frac{n}{2}) + cn$

$\Rightarrow T(n) = O(n^{\log_2 4}) = O(n^2)$

Still quadratic time complexity.

So not more efficient than traditional algo.

— A more efficient recursive algorithm.

Consider

$$\underbrace{(\underbrace{x_1}_{n/2} + \underbrace{x_0}_{n/2})}_{n/2}\underbrace{(y_1 + y_0)}_{n/2} = \underbrace{x_1 y_1}_{A} + \underbrace{x_1 y_0}_{B} + \underbrace{x_0 y_1}_{C} + \underbrace{x_0 y_0}_{D}$$

$$\underbrace{x_1 y_0 + x_0 y_1}_{B+C} = \underbrace{\underbrace{(x_1 + x_0)}_{n/2}\underbrace{(y_1 + y_0)}_{n/2}}_{E} - \underbrace{\underbrace{x_1}_{n/2}\underbrace{y_1}_{n/2}}_{A} - \underbrace{\underbrace{x_0}_{n/2}\underbrace{y_0}_{n/2}}_{D}$$

RecursiveMultiply_($x, y, b$)

n = # digits in x
if $n < 2$
  return $x * y$
else
  $x_1$ = first $\frac{n}{2}$ digits of $x$
  $x_0$ = last $\frac{n}{2}$ digits of $x$
  $y_1$ = first $\frac{n}{2}$ digits of $y$
  $y_0$ = last $\frac{n}{2}$ digits of $y$
  Compute $(x_1 + x_0)$ and $(y_1 + y_0)$
  E = RecursiveMultiply $(x_1 + x_0, y_1 + y_0, b)$
  A = RecursiveMultiply $(x_1, y_1, b)$
  D = RecursiveMultiply $(x_0, y_0, b)$
  } 3 recursive splits
  return $x_1 y_1 \times b^n + (E - A - D) \times b^{n/2} + x_0 y_0$

Therefore, using 3 splits only

$$T(n) = O(n^{\log_2 3}) = O(n^{1.59})$$