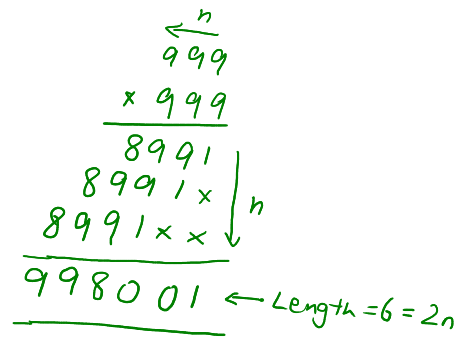


- We need to multiply 2 n-digit (or n-bit) numbers.
- Answer can be 2n digits long.
- Traditional algorithm is efficient ($O(n^2)$).

n^2 multiplications plus upto $n^2 + n - 2$ additions.
So $O(n^2)$ effort.

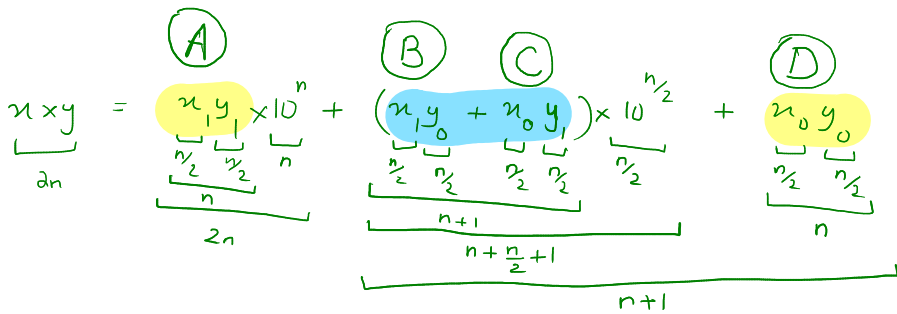


$$2(n-1 + \dots + 1 + 0) + 2n - 2 = (n-1)n + 2n - 2 = n^2 + n - 2 \text{ additions at max.}$$

- A recursive algorithm.

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

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



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 $A \times b^n + (B + C) \times b^{n/2} + D$

4 mults of $n/2$ digit numbers

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

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

4 multiplications of $n/2$ digit numbers
 $T(n) \leq 4T(n/2) + cn$
 $\Rightarrow T(n) = O(n^{\log_2 4}) = O(n^2)$

Still quadratic time complexity.

So not more efficient than traditional algo.

- A more efficient recursive algorithm.

Consider

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

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

RecursiveMultiply(x, y, b)

n = # digits in x

if n < 2

return x * y

else

x₁ = first n/2 digits of x

x₀ = last n/2 digits of x

y₁ = first n/2 digits of y

y₀ = last n/2 digits of y

Compute (x₁ + x₀) and (y₁ + y₀)

E = RecursiveMultiply(x₁ + x₀, y₁ + y₀, b)
 A = RecursiveMultiply(x₁, y₁, b)
 D = RecursiveMultiply(x₀, y₀, b)

} 3 recursive splits

return x₁ y₁ × bⁿ + (E - A - D) × b^{n/2} + x₀ y₀

Therefore, using 3 splits only

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