CompSci 161
Winter 2023 Lecture 6:
Divide and Conquer III:
Multiplication Algorithms

PSet 2 due Wed
PSet 3 release Wed
due NOT wed. (why not?)

2

Integer Multiplication

- Given two n-bit integers X, Y, compute $X \cdot Y$
- Example: What is $13 \cdot 11$?

$$\begin{array}{r} 13 \\ \times\ 11 \\ \hline 13 \\ 13\text{X} \\ \hline 143 \end{array}$$

3
What is a Computer Anyway?

- Example: What is $13 \cdot 11$?

13	11	
26	5	
~~52~~	~~2~~	even
104	1	

99	99	
198	49	
~~396~~	~~24~~	even
~~792~~	~~12~~	
~~1584~~	~~6~~	
3168	3	
6336	1	
9801		

4

Al-Khwarizmi's algorithm

Why does al-Khwarizmi's algorithm work?

- Example: What is $13 \cdot 11$?

```
 13
x11
---
```

```
        1101     13
     x  1011     11
    ------------
         1101    13
        1101X    26
       0000XX
      1101XXX    104
    ------------
     10001111    143
```

5

Starting D&C for Integer Multiplication

$$X \times Y = (X_H \times 2^{n/2} + X_L) \times (Y_H \times 2^{n/2} + Y_L)$$
$$= X_H \cdot Y_H \times 2^n + (X_H Y_L + X_L Y_H) \times 2^{n/2} + X_L Y_L$$

$X = 13 = 1101$ $\quad X_H = 11 = 3$ $\quad X_L = 01 =$

Algorithm Mult(X, Y)

Create X_H, X_L, Y_H, Y_L

A = Mult(X_H, Y_H)

B = Mult(X_H, Y_L)

C = Mult(X_L Y_H)

D = Mult(X_L Y_L)

That is $\Theta(n^2)$

$$T(n) = 4T(n/2) + \Theta(n)$$

Return $A \cdot 2^n + (B + C) \cdot 2^{n/2} + D$

$\times 2^k$ is a bitshift

6 ## Improved Integer Multiplication

$(X_H + X_L)(Y_H + Y_L) = X_H Y_H + X_H Y_L + X_L Y_H + X_L Y_L$

- Do we really need to compute $X_H \cdot Y_L$?

$A = \text{mult}(X_H \; Y_H)$

$D = \text{mult}(X_L \; Y_L)$

$E = \text{mult}(X_H + X_L, \; Y_H + Y_L)$

$G = E - A - D \quad // = X_H Y_L + X_L Y_H$

return $A \cdot 2^n + G \cdot 2^{n/2} + D$

$T(n) = 3T\left(\frac{n}{2}\right) + \Theta(n)$

is $\Theta(n^{\log_2 3})$

7 ## Integer Multiplication Example

- Example: What is $13 \cdot 11$?

X_H	X_L	Y_H	Y_L
11	01	10	11
3	1	2	3

8 ## Matrix Multiplication

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} I & J \\ K & L \end{bmatrix}$$

- $I = AE + BG$
- $J = AF + BH$
- $K = CE + DG$
- $L = CF + DH$

$T(n) = 8T(\frac{n}{2}) + n^2$

- Running time if basis for D & C algorithm?
- [redacted]

See Strassen's in notes