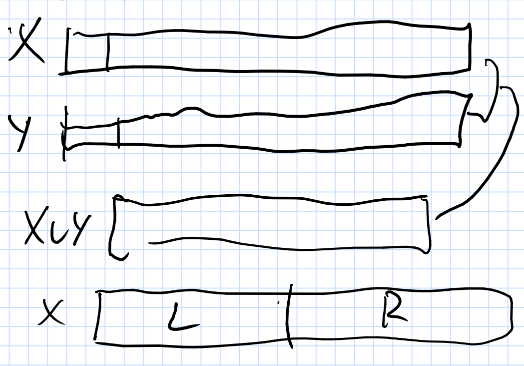
374: Divide and Conquer

9/16/21

Lecture 8

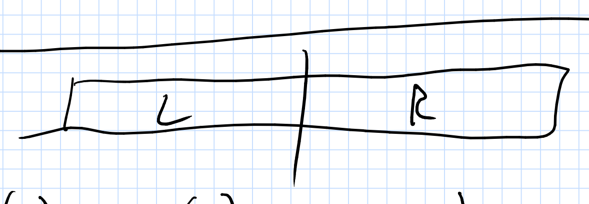
merge sort
radix sort
multiplying things:
- Complex numbers
- Large numbers
- Matrices (Strassen)
- Maximum subarray
- Closest pair (?)

Merge Sort



$X[1 \ldots n]$ $O(n^2)$

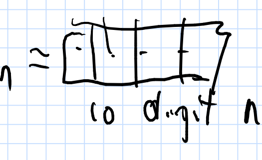
in place merge.



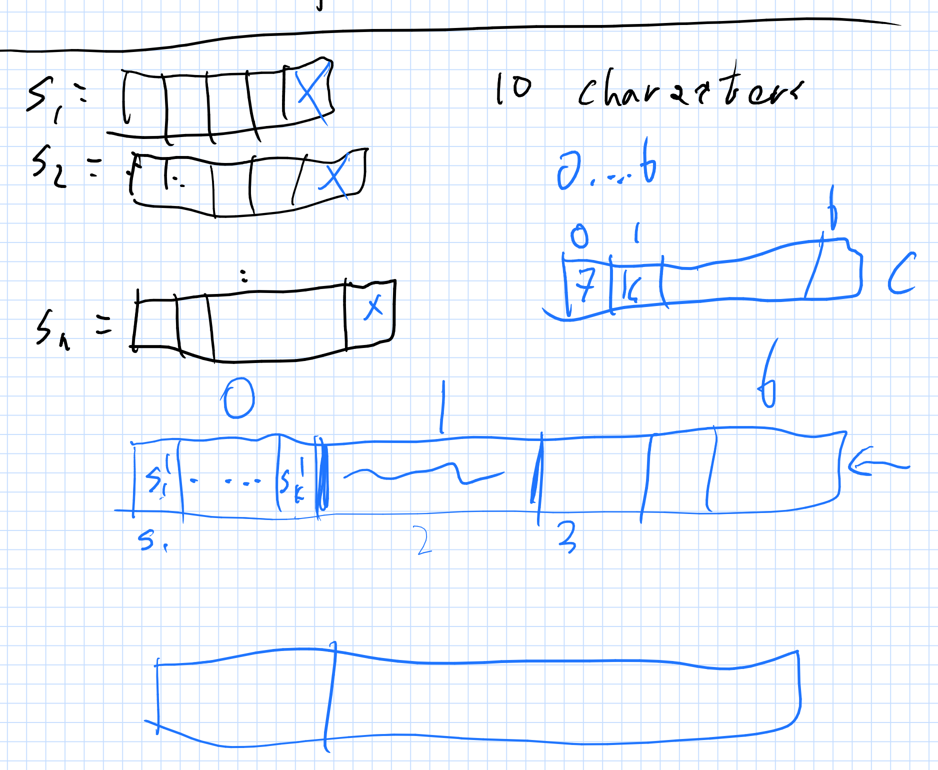
$T(n) = O(n) + 2T(\frac{n}{2})$ Merge sort
$= O(n \log n)$.

Radix sort

$x_1, x_2, \ldots, x_n$ integer numbers $\{1, \ldots n^{10}\}$



$x_n =$ 10 digit number



10 characters
0…b

$O(n \cdot w)$
w: # of digits in each number

$1 \ldots n^{11}$ n $O(n)$
w = 10 $O(n)$

Multiplying things

- Egyptian multiplication 3700 years
  X and Y multiply them

	X	Y	
X is odd	X − 1	Y	Y
X is even	X/2	2Y	—
Y = 1			Y

Gauss multiplying complex number

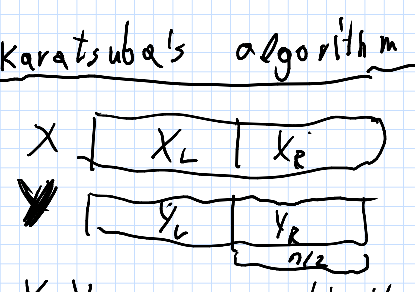
$(\alpha + \beta i)(\alpha' + \beta' i)$
$= \alpha\alpha' + \alpha\beta' i + \beta\alpha' i - \beta\beta'$ 4 multiplications

$(\alpha + \beta)(\alpha' + \beta') = \alpha\alpha' + \alpha\beta' + \beta\alpha' + \beta\beta'$
$\alpha\beta' + \beta\alpha'$

$(\alpha + \beta i)(\alpha' + \beta' i)$
$= (\alpha\alpha' - \beta\beta')$
$+ (\delta - \alpha\alpha' - \beta\beta')i$

3 mult.
$\delta = (\alpha + \beta)(\alpha' + \beta')$

Karatsuba's algorithm



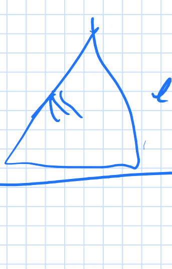
n digits
n is a power of 2

$X \cdot Y \approx 2n$ digits $O(n^2)$

$X \cdot Y = 10^n \cdot X_L \cdot Y_L + 10^{n/2}(X_L Y_R + X_R Y_L) + X_R Y_R$

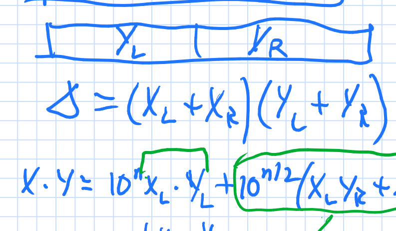
multiplying n digits $\Rightarrow$ 4 mult of n/2 digits

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



$\log_2 n$

n^2



$\delta = (X_L + X_R)(Y_L + Y_R)$

$X \cdot Y = 10^n X_L \cdot Y_L + 10^{n/2}(X_L Y_R + X_R Y_L) + X_R \cdot Y_R$

$10^{n/2}(\delta - X_L Y_L - X_R Y_R)$

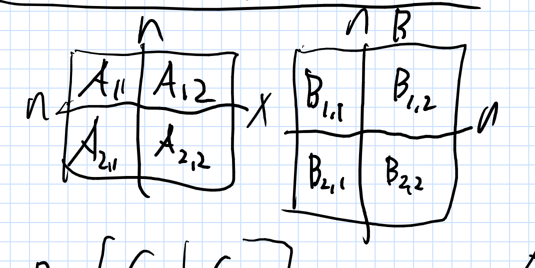
$a = X_L Y_L$
$b = X_R Y_R$
$\delta = (X_L + X_R)(Y_L + Y_R)$

$x \cdot y = 10^n a + (\delta - a - b) \cdot 10^{n/2} + b$
$O(n)$ $O(n)$

$T(n) = O(n) + 3T(\frac{n}{2})$
$h = \log_2 n$
$3^h = (2^{\log_2 3})^{\log_2 n} = n^{\log_2 3}$

$O(n^{\log_2 3}) \lll O(n^2)$
$O(n^{1.585\ldots})$

Strassen algorithm



$O(n^3)$
$A, B: n \times n$
$A_{11} \ldots A_{22}, B_{11}, \ldots, B_{22}: n/2 \times n/2$

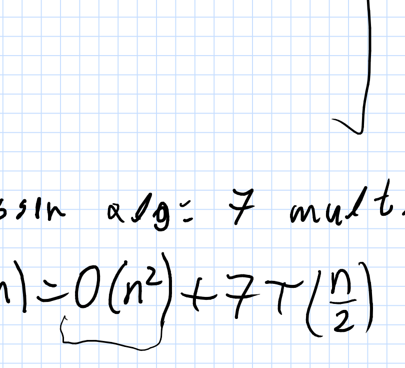
$$A \cdot B = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

$C_{11} = A_{11} \cdot B_{11} + A_{12} \cdot B_{21}$
C_2
C_3
C_4

8 products of $n/2 \times n/2$ matrices

Strassen alg: 7 multiplications

$T(n) = O(n^2) + 7T(\frac{n}{2})$



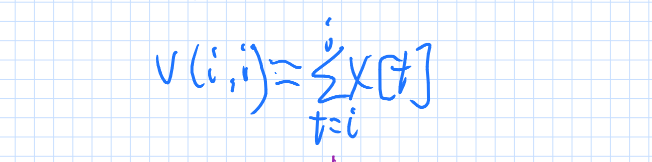
7^i
$h = \log_2 n$
$\frac{n}{2^i} \times \frac{n}{2^i}$

$$\sum_{i=0}^{h} O\left(\left(\frac{n}{2^i}\right)^2 \cdot 7^i\right) = O\left(n^2 \sum_{i=0}^{h} \left(\frac{7}{4}\right)^i\right)$$
$$= O\left(n^2 \cdot \left(\frac{7}{4}\right)^{\log_2 n}\right)$$
$$= O\left(n^{2 + \log_2 \frac{7}{4}}\right) << O(n^3)$$
$$\approx O(n^{1.85\ldots})$$

$O(n^\omega)$: Fastest possible alg for multiplying $n \times n$ matrices

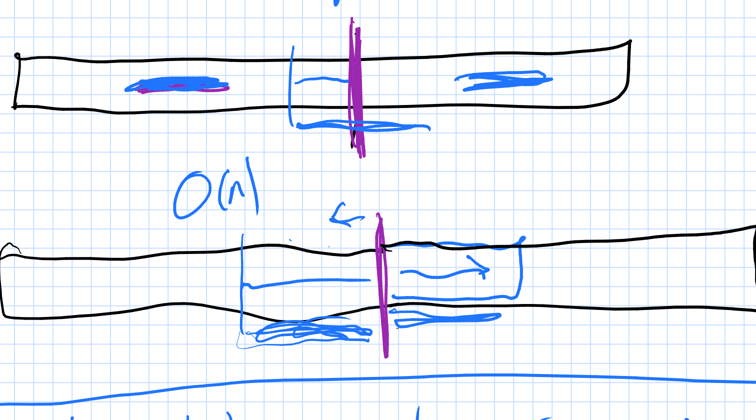
$\omega \le 2.37287$

Maximum subarray



X n number

$V(i, j) = \sum_{t=i}^{j} X[t]$



$O(n)$

$T(n) = O(n) + 2T(\frac{n}{2}) = O(n \log n)$.