classmate
Date
Page

★ GCD of a, b : $\gcd(a,b)$

$a > b \geq 1$

★ $S: \{ax + by \mid x, y \in \mathbb{Z}\}$

$d = \gcd(a,b)$

Let s be the min element of S

∴ By WOP:

$s = ax_1 + by_1$

$d \mid a$ and $d \mid b$ ∴ $d \mid s \Rightarrow d \leq s$

$a = sq + r$

$\therefore r = a - sq$

$= a(1 - qx_1) + b(-qy_1)$

WOP (∵ $r < s$)

∴ $r = 0 \Rightarrow s \mid a$!!! $s \mid b$

∴ $s \mid d$ ∴ $s \leq d$

$\therefore d = s$

★ GCD recursion Theorem

$\underbrace{\gcd(a,b)}_{d} = \underbrace{\gcd(b, a \bmod b)}_{d_1}$

$d \mid d_1$ & $d_1 \mid d$

∴ $d = d_1$

Proof: Use $a = bq + a \bmod b$

* Euclid's GCD Algorithm

```
int gcd (int a, int b)
{
    if (b == 0) return a;
    return gcd (b, a%b);
}
```

→ Correctness : GCD recursion theorem

→ Terminates : b always ↓

* Tail recursive

k: iterations

k = f(a,b)

$r_1 = b \overline{)\, a \,(} q_1$

$r_2 \overline{)\, r_1 \,(} q_2$

$r_3 \overline{)\, r_2 \,(} q_3$

⋮

$r_k \overline{)\, r_{k-1} \,(} q_k$

0

* Induction on k :

T.P.T. $a \geq f_{k+2}$, $b \geq f_{k+1}$ (Induction hypo)

1	1	2	3
f_1	f_2	f_3	f_4

* Induction Base : k = 1

$b \geq 1$ & $a \geq 2$ $\because a > b$

$\therefore b \geq f_2$ & $a \geq f_3$

Hence Base ✓

* Induction Step:

Let the no. of iterations be k-1 $\therefore r_1 \geq f_{(k-1)+2}$

$\therefore b \geq f_{(k-1)+2}$ $\left.\vphantom{\therefore b \geq f_{(k-1)+2}}\right\}$ $\because b = r_1$ ↓ then $r_1 \approx a$ $r_2 \geq f_{(k-1)+1}$

$\Rightarrow b \geq f_{k+1}$ $r_2 \approx b$

But $b = r_1$

$\therefore r_1 \geq f_{k+1} \Rightarrow b \geq f_{k+1}$

$\because a = bq + r_2$

$\geq b + r_2 \geq f_{k+1} + f_k \geq f_{k+2}$

$\therefore b \geq f_{k+1} \geq \phi^{k-1}$

$\therefore k$ is $O(\log b)$

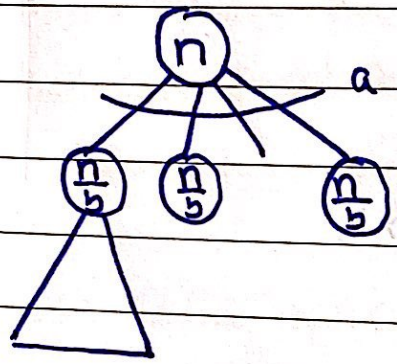
* Weakly polynomial Time algorithm

* Strongly polynomial Time algorithm :

TC based on no. of inputs like sorting.

★ Divide and Conquer

★ Reccurance Tree



★ b and a maybe different

★ $T(n) = aT(n/b) + \underbrace{f_1(n)}_{\text{Divide}} + \underbrace{f_2(n)}_{\text{Conquer}}$ $n > c$

$= f_3(n)$ $n \leq c$

→ Same subproblem doesnt occur at any other node

★ $T(n) = aT(n/b) + f(n)$ $n > c$

$= O(1)$ $n \leq c$

→ Master Theorem:

① If $f(n) = O(n^{\log_b a - \epsilon})$ $\epsilon > 0$

then $T(n) = \Theta(n^{\log_b a})$

② If $f(n) = \Theta(n^{\log_b a})$

then $T(n) = \Theta(n^{\log_b a} \log n)$

③ If $f(n) = \Omega(n^{\log_b a + \epsilon})$ & $af(n/b) \leq cf(n)$ $c<1$

then $T(n) = \Theta(f(n))$

classmate
Date
Page

* $T(n) \leq qT(n/2) + c_1 n$ for $n > 2$
  $\leq c_2$ for $n \leq 2$

① If $q = 1$
  ③ follows $\therefore T(n) = \Theta(n)$

② If $q = 2$
  ② follows $\therefore T(n) = \Theta(n \log n)$

③ If $q > 2$
  ① follows $\therefore T(n) = \Theta(n^{\log_2 q})$

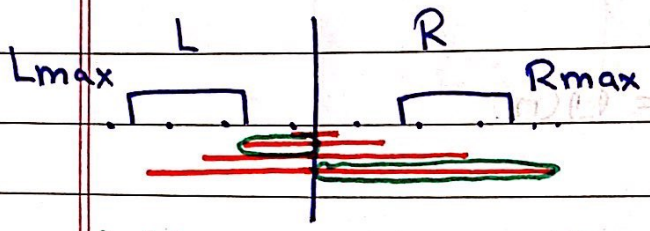
* Correctness of Divide and Conquer
Induction Base: Leaf node
Induction Hypo: Merge two subtrees to give correct result.

* Divide and Conquer used to optimise polynomial time algorithm.

① Maximum Subarray

Naive: $O(n^3)$: $n(n+1)/2$ subarrays

$\Sigma[l,r]$ takes $O(n)$



* Max Sum across the splitter

$T(n) = 2T(n/2) + \underbrace{O(1)}_{\text{Divide}} + \underbrace{O(n)}_{\text{Conquer}}$ if $n > 1$

$= O(1)$ $n = 1$

Prefix & Suffix Sums

$\therefore T(n) = \Theta(n \log n)$

* Correctness:

Induction on no. of elements

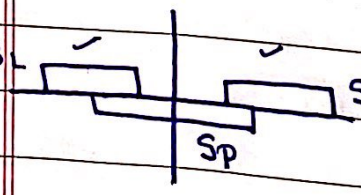
* Induction Basis:

$n = 1$: Output element itself

* Induction Hypo:

If no. of elements $< n$ the algo outputs correctly

* Induction Step: no. of elements $= n$



Take max $\{S_L, S_P, S_R\}$

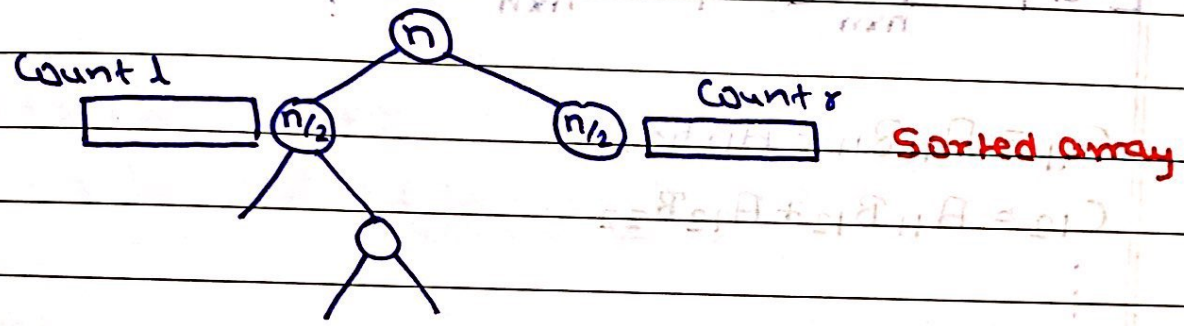
Subarray entirely in left or right or across the splitter

② Counting Inversions:

$a_1\ a_2 \ldots a_i \ldots a_j \ldots a_n$

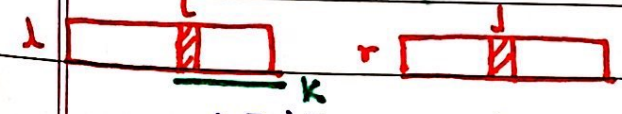
$a_i > a_j \rightarrow$ 1 inversion

Naive: $O(n^2)$



Inversion count = count l + count r + count m

↑ inversions while merging



l[i] > r[j] (# Sorted array)

then count m += k

→ $T(n) = 2T(n/2) + O(1) + O(n)$ if $n > 1$

$= O(1)$ if $n = 1$

$\therefore T(n) = O(n \log n)$

③ ★ Matrix Multiplication

Naive $O(n^3)$

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}_{n \times n} \quad \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}_{n \times n} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

$C_{11} = A_{11}B_{11} + A_{12}B_{21}$

$C_{12} = A_{11}B_{12} + A_{12}B_{22}$

⋮

$\therefore T(n) = 8T(n/2) + O(1) + O(n^2) \quad n > 1$

$= O(1) \quad n = 1$

Divide Conquer # Addition

★ Strassen's Matrix Multiplication

$C_{12} = \underbrace{A_{11}(B_{12} - B_{22})}_{P_1} + \underbrace{(A_{11} + A_{12})B_{22}}_{P_2}$

$C_{21} = \underbrace{(A_{21} + A_{22})B_{11}}_{P_3} + \underbrace{A_{22}(B_{21} - B_{11})}_{P_4}$

$C_{11} = \underbrace{(A_{11} + A_{22})(B_{11} + B_{22})}_{P_5} + P_4 - P_2$

$+ \underbrace{(A_{12} - A_{22})(B_{21} + B_{22})}_{P_6}$

$C_{22} = P_5 + P_1 - P_3 - \underbrace{(A_{11} - A_{21})(B_{11} + B_{12})}_{P_7}$

$T(n) = 7T(n/2) + O(1) + O(n^2) \quad n > 1$

$= O(1) \quad n = 1$

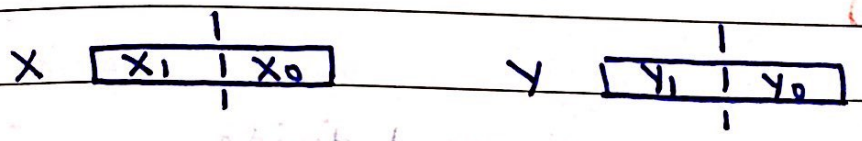
$\therefore$ TC $O(n^{\log_2 7}) = O(n^{2.81})$

→ Read Coppersmith – Winograd's Algo.

⑤ Karatsuba's Algo:

Multiply two n bit positive.

→ RAM model, bitwise logical op & right and left shift



$$XY = (X_1 \cdot 2^{n/2} + X_0)(Y_1 \cdot 2^{n/2} + Y_0)$$

$$= (X_1Y_1 \cdot 2^n + (X_0Y_1 + X_1Y_0)2^{n/2} + X_0Y_0) \quad \text{①}$$

$$T(n) = 4T(n/2) + \underbrace{O(1) + O(n)}_{\text{divide, conquer}} \quad n > 1$$

$$= O(1) \quad n = 1$$

$$\therefore T(n) = O(n^2)$$

① $\equiv (X_1 + X_0)(Y_1 + Y_0) - X_1Y_1 - X_0Y_0$

$$\therefore T(n) = 3T(n/2) + O(1) + O(n) \quad n > 1$$

$$= O(1) \quad n = 1$$

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

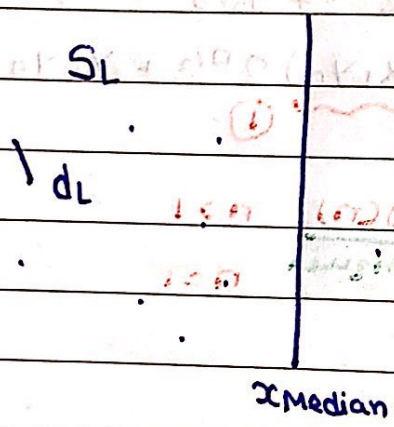
(5) Closest Pair of Points (CPP)

No two have same x or same y

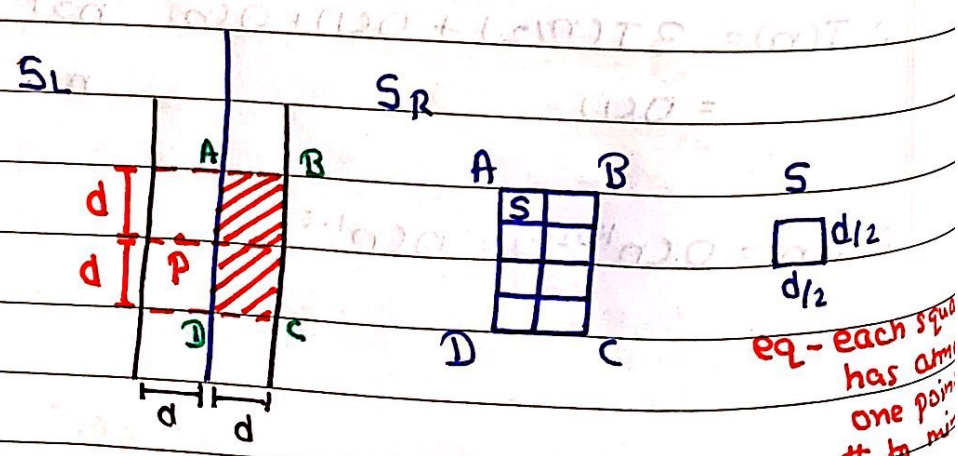
every 2 points have distinct distances

Naive: $O(n^2)$

Find median x - co-ordinate & divide



$d = \min(d_L, d_R)$



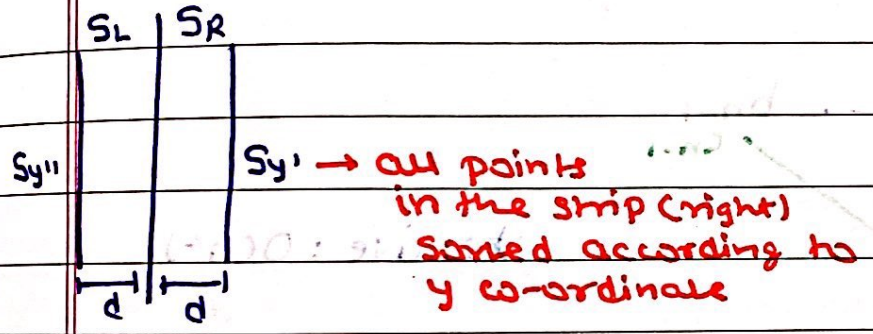
eg - each square has atmost one point # to min of d

★ We need to check only the points (pairs) with one $\in S_L$ and other $\in S_R$

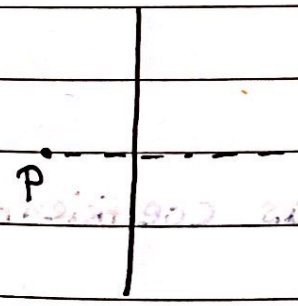
classmate
Date
Page

★ Hence for each point P in the strip Sy'' we need to check 4 points above and below it in Sy'

Sx: Sorted according to x-coordinate
Sy: Sorted according to y-coordinate
} Pre Compute



Sy' → all points in the strip (right) Sorted according to y co-ordinate

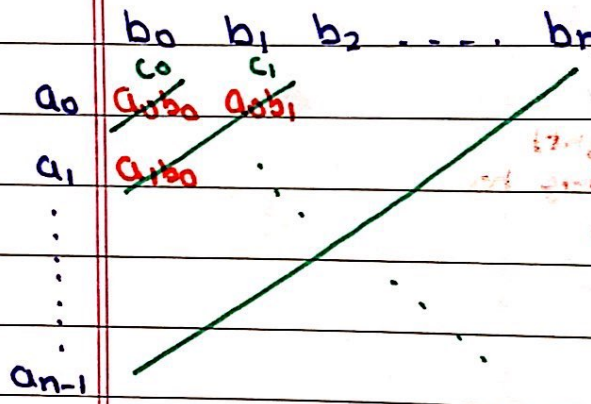


⑥ Fast Fourier Transform (FFT)

$$A(x) = \sum_{i=0}^{n-1} a_i x^i \quad , \quad B(x) = \sum_{i=0}^{n-1} b_i x^i$$

$$C(x) = A(x) \cdot B(x)$$

$$= c_0 + c_1 x + \ldots + c_{2n-2} x^{2n-2}$$



★ Naive : $O(n^2)$

★ Polynomial can be represented as coefficient representation.

Vector of coefficients

★ Set of points representation

$A(x) = \{(x_0, y_0), (x_1, y_1), \ldots (x_{n-1}, y_{n-1})\}$

$y_k = A(x_k)$

★ Horner's Rule :

$A(x) = a_0 + \ldots x(a_{n-3} + x(a_{n-2} + x a_{n-1}))$

↓

Takes $O(n)$

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^{n-1} \\ 1 & x_1 & \cdots & \cdots & \cdots \\ \vdots & & & & \\ 1 & x_{n-1} & \cdots & \cdots & x_{n-1}^{n-1} \end{pmatrix} = X \qquad \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = A$$

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} = Y$$

$* \; XA = Y$

$\therefore A = X^{-1}Y$

* Deg(C(x)) is $2n-2$

Hence atleast $2n-1$ points required to find its coefficients.

→ Step 1:

$x_0, x_1, \dots, x_{2n-2}$

Find $A(x_i)$ & $B(x_i)$ for each x_i : $O(n^2)$

→ Step 2:

$C(x_k) = A(x_k) \cdot B(x_k)$: $O(n)$ $(\because 0 \le k \le 2n-2)$

→ Step 3:

Find c.e. of C(x)

$C = X^{-1}Y$: $\Omega(n^2)$

$A(x) = a_0 + a_1 x + \ldots + a_{n-1} x^{n-1}$

$A_{even}(x) = (a_0 + a_2 x + a_4 x^2 + a_{n-2} x^{n/2-1})$

$A_{odd}(x) = (a_1 + a_3 x + a_5 x^2 + \ldots + a_{n-1} x^{n/2-1})$

$\therefore A(x) = A_{even}(x^2) + x A_{odd}(x^2)$

* Note Here we have to evaluate $A(x)$ of deg $n-1$ at $2n$ points Let the time required to do so be $T(n)$

→ Twiddle Factors:

$w_{j,r} = e^{2\pi j/r \cdot i}$

$= (\cos 2\pi j/r, \sin 2\pi j/r)$

→ Let the required $2n$ points be

$w_{0,2n}, w_{1,2n}, \ldots, w_{2n-1,2n}$

* $A(w_{j,2n}) = A_{even}(w_{j,2n}^2) + w_{j,2n} A_{odd}(w_{j,2n}^2)$

$w_{j+n,2n} = e^{2\pi(j+n)/2n \cdot i}$

$= e^{2\pi j/2n \cdot i} \cdot e^{i\pi}$

$= -w_{j,2n}$

$\therefore w_{j+n,2n}^2 = w_{j,2n}^2$ *

$A(j) = A_{even}(j^2) + j\,A_{odd}(j^2)$

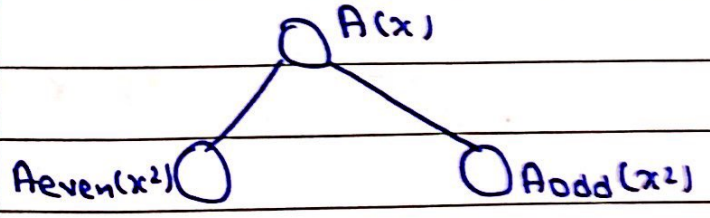
$A(-j) = A_{even}(j^2) - j\,A_{odd}(j^2)$

$A(w_{j,2n}) = A_{even}(w_{j,2n}^2) + w_{j,2n}\,A_{odd}(w_{j,2n}^2)$ ... (1)

(#*) $A(w_{j+n,2n}) = A_{even}(w_{j,2n}^2) - w_{j,2n}\,A_{odd}(w_{j,2n}^2)$ ... (2)

$A(w_{j,2n})$ <
- $j = 0, 1, \ldots, n-1$ Apply (1)
- $j = n, \ldots, 2n-1$ Apply (2)

* So To calculate $A(x)$ for $2n$ values we effectively need to calculate $A_{even}(x^2)$ & $A_{odd}(x^2)$ for only n values



$\therefore T(n) = 2T(n/2) + O(n) + O(n)$
Divide Combine

$\therefore T(n) = O(n \log n)$ for step 1

* DFT: Discrete Fourier Transform
Represent a polynomial using Twiddle Factors