$$\begin{array}{cc} x_1 & x_2 \end{array}$$

$$x = \boxed{1\,2\,3}\boxed{4\,5\,6} \qquad x = x_1 \cdot 10^m + x_2$$

$$y = \boxed{7\,8\,7}\boxed{1\,1\,1} \qquad y = y_1 \cdot 10^m + y_2$$

$$\begin{array}{cc} y_1 & y_2 \end{array}$$

n

$m = n/2$

$$x \cdot y = (x_1 \cdot 10^m + x_2)(y_1 \cdot 10^m + y_2)$$

$$T(n) = 4 \cdot T\left(\tfrac{n}{2}\right) + O(n)$$

$$= x_1 y_1 \cdot 10^{2m} + (x_2 y_1 + x_1 y_2) \cdot 10^m + x_2 y_2$$

$$= x_1 y_1 \cdot 10^{2m} + (x_1 + x_2)(y_1 + y_2) \cdot 10^m + x_2 y_2 - x_1 y_1 \cdot 10^m - x_2 y_2 \cdot 10^m$$

$$(x_1 + x_2)(y_1 + y_2)$$

$$= x_1 y_1 + x_2 y_1 + x_1 y_2 + x_2 y_2$$

$$= x_1 y_1 (10^{2m} - 10^m) + (x_1 + x_2)(y_1 + y_2) \cdot 10^m + x_2 y_2 (1 - 10^m)$$

$$T(n) = 3T\left(\tfrac{n}{2}\right) + O(n)$$

$$n^{\log_b a} = n^{\log_2 3} < n^{1.59}$$

Karatsuba algorithm

$\therefore O(n^{1.59})$

Ex.

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

$$q(x) = b_0 + b_1 x + \cdots \cdots + b_n x^n$$

Exercise

Matrix Multiplication

A, B are $n \times n$ matrices

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$$

$$\downarrow$$

$C = A \times B$

$$C_{i,j} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj} \qquad O(n^3)$$

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \quad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

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

$C_{12} =$ – — —

$C_{21} =$ – – —

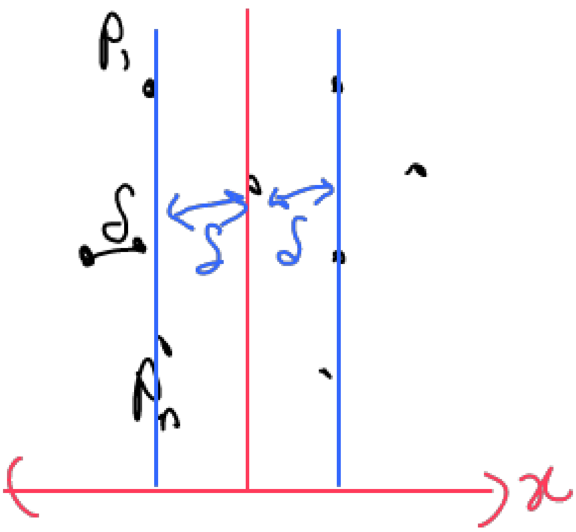
$C_{22} =$ – — —

$$T(n) = 8\,T\left(\frac{n}{2}\right) + O(n^2)$$

Strassen Algorithm

Refer to course notes

Closest Pair (A)



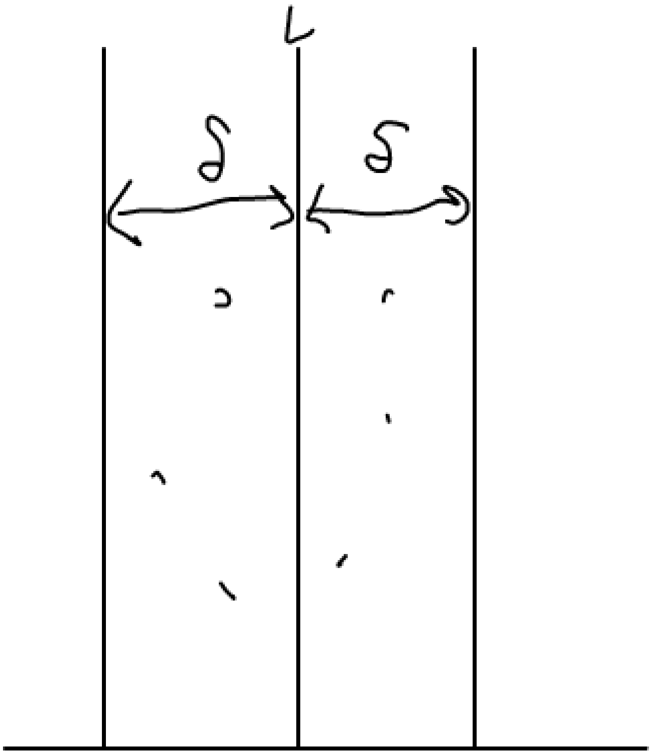
for $i = 1 \ldots n$
  for $j = 1 \ldots n$
    compute $d(i, j)$

$p = (p.x, p.y)$

$$\text{closest}(A) = \min \begin{cases} \text{closest}(Q) \\ \text{closest}(R) \\ \vdots \\ \underline{\text{closest}(Q, R, \delta)} \end{cases}$$

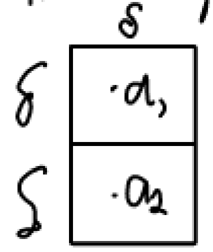
let $\delta = \min(\delta_1, \delta_2)$

for each $p \in Q$
  for each $q \in R$
    $d(p, q)$

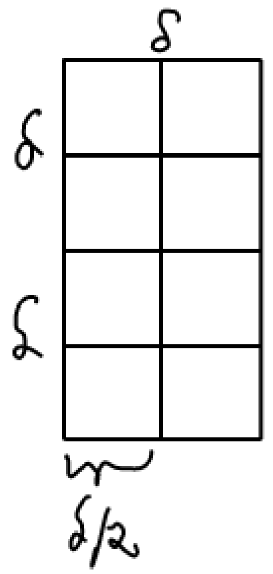


for $p \in$ left

How many?



$d(q_1, q_2) \geq \delta$



Lemma: there are at most one point in each $\delta/2 \times \delta/2$ block

proof: a pair of points have distance $\leq$ diameter $= \frac{\delta}{\sqrt{2}} < \delta$

so, contradiction.

$\therefore$ compare at most 8 points in right.

$$T(n) = 2 \cdot T\left(\frac{n}{2}\right) + O(n)$$
$$= O(n \log n)$$

① divide A into $Q \cup R$

② closest$(Q, Q_x, Q_y) \to \delta_1$

③ closest$(R, R_x, R_y) \to \delta_2$

④ $\delta = \min(\delta_1, \delta_2)$

⑤ closest(Q, R, δ)

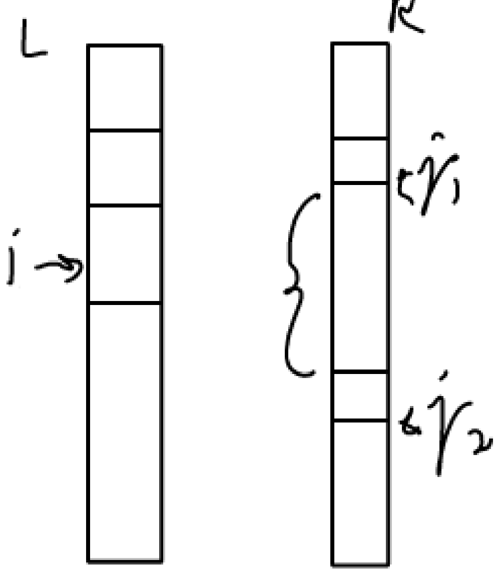
$$T(n) = 2 \cdot T\left(\frac{n}{2}\right) + O(n)$$
$$= O(n \cdot \log n)$$

array sorted by y-axis

A	A_x	A_y
original	sorted by x	sorted by y

$\therefore$ closest(A, A_x, A_y)

```
right = {}
for i = 1 .. n
    if (A_y[i].x ∈ [L, L+δ])
        right = right + A_y[i]
```



$$j_1 = \max\{j \mid R[j_1] < L[i] - \delta\}$$

$$j_2 = \max\{j \mid R[j_2] > L[i] + \delta\}$$

$j_1 \leftarrow 0$, $j_2 \leftarrow 1$

for $i = 1 .. |left|$

  while($R[j_1 + 1] < L[i] - \delta$)

    j_1++;

  while($R[j_2] \leq L[i] + \delta$)

    j_2++;

  compare $L[i]$ vs $R[j_1 + 1 .. j_2 - 1]$