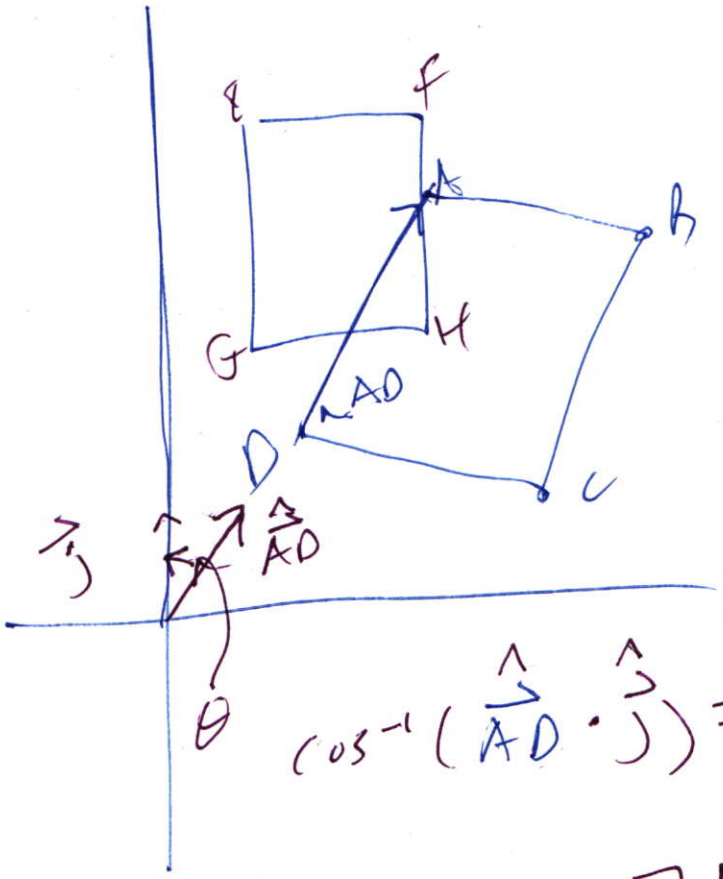
Slide 2



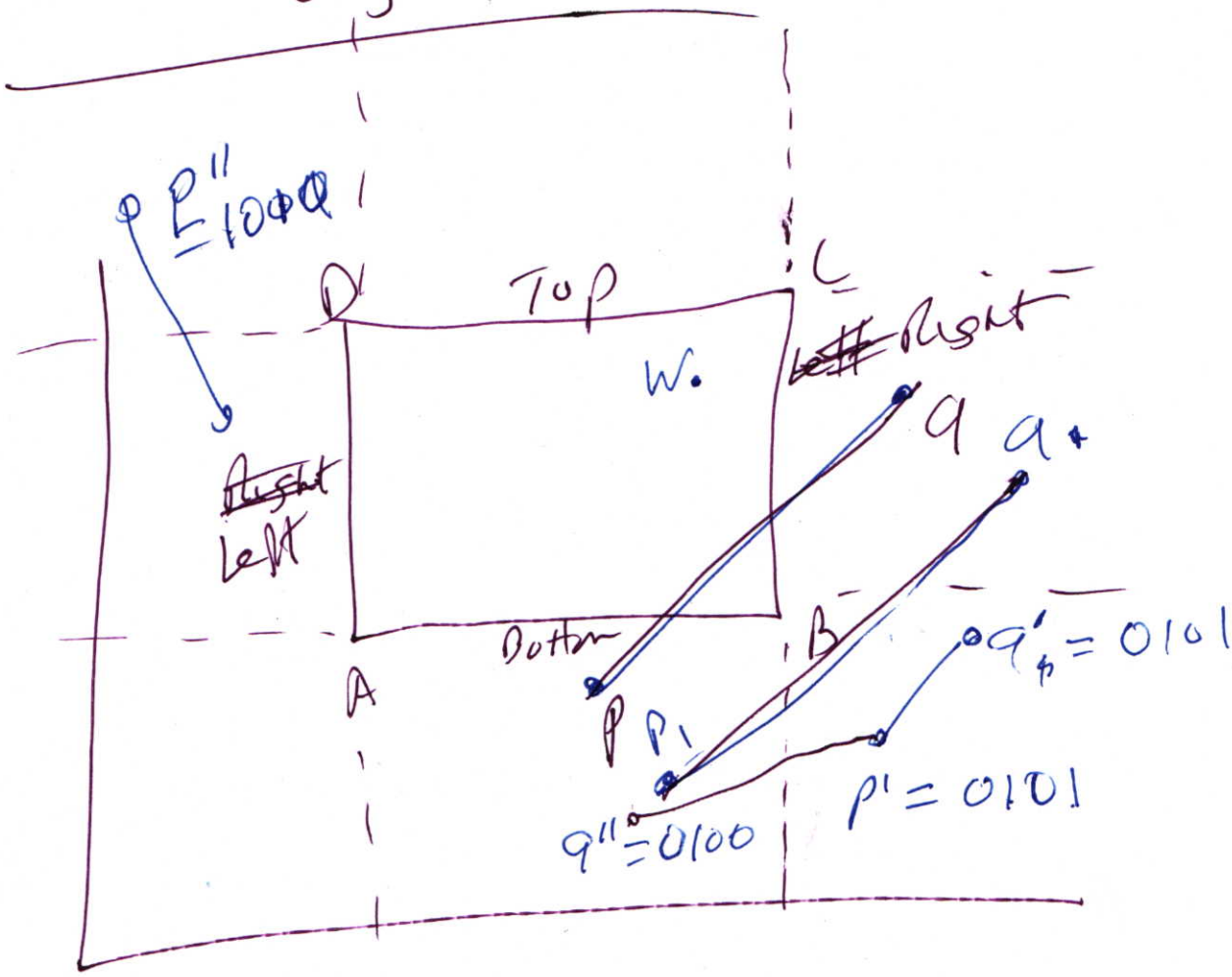
$\vec{j} = \langle 0, 1 \rangle$

$\cos^{-1}(\hat{AD} \cdot \hat{j}) = \theta$

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} A_x \\ A_y \end{bmatrix}$$

Rotated pts $= M \begin{bmatrix} A_x & B_x & C_x & D_x & \cdots \\ A_y & B_y & C_y & D_y \end{bmatrix}$

Cohen Sutherland algorithm



Px TBLR

P = 1000

q = 0001

w = 0000

left = $C_x = B_x$

right = $A_x = D_x$

top = $D_y = C_y$

bottm = $A_y = B_y$

Slide 3

$p = 0100$
$q = 0001$

$p' = 0101$
$q' = 0101$
$q'' = 0100$
$p'' = 1000$

p, q
bp, bq

bp and bq has one on bit
$\Rightarrow$ reject

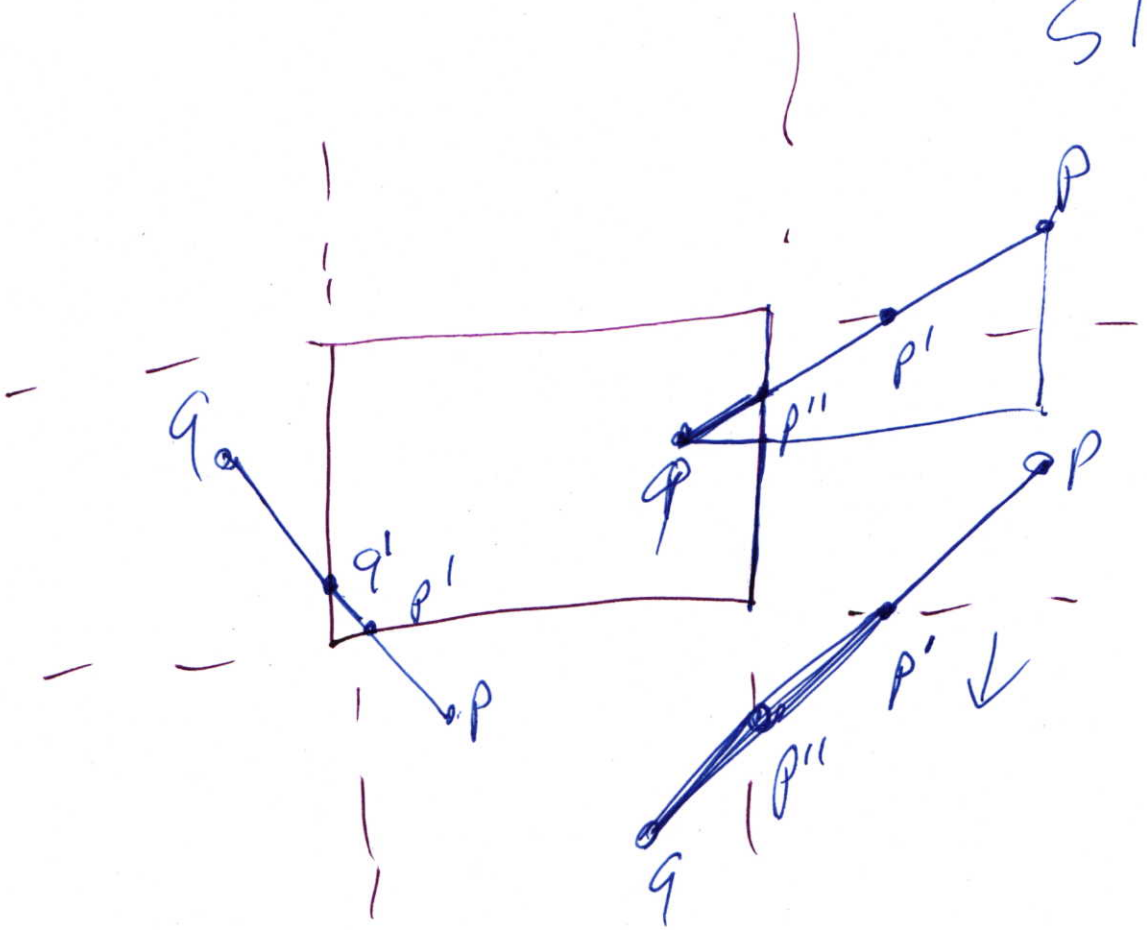
$bp \;\&\; bq \neq 0$

if bq = code p
  hq = code q
  if hq = 0 and hp = 0 ⇒ accept
else if hq and hq == 0
  reject

else
  set of p is all zero
    switch
      [p is outside
        q is maybe inside

  clip p to boundary
  repeat

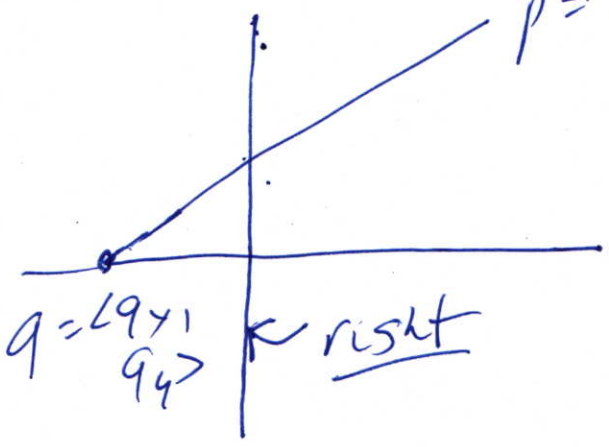
Slide 5



$q = \langle q_x, q_y \rangle$

$p = \langle p_x, p_y \rangle$

$v = p - q = \langle \Delta x, \Delta y \rangle$

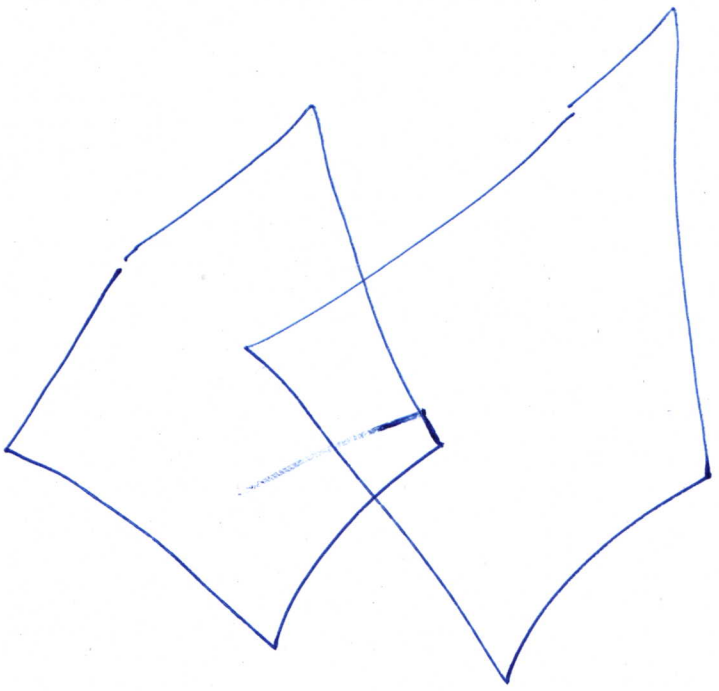


$y = mx + b$

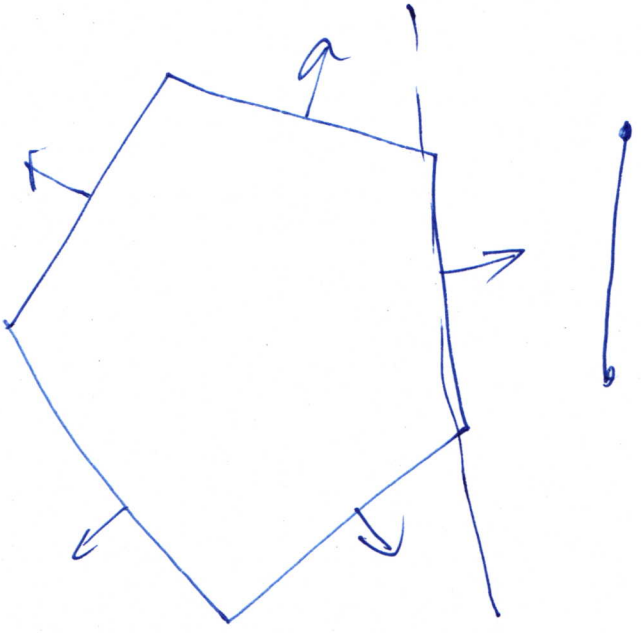
$y = \frac{\Delta y}{\Delta x}_{right} + q_y$

$x = right$

Slide 6

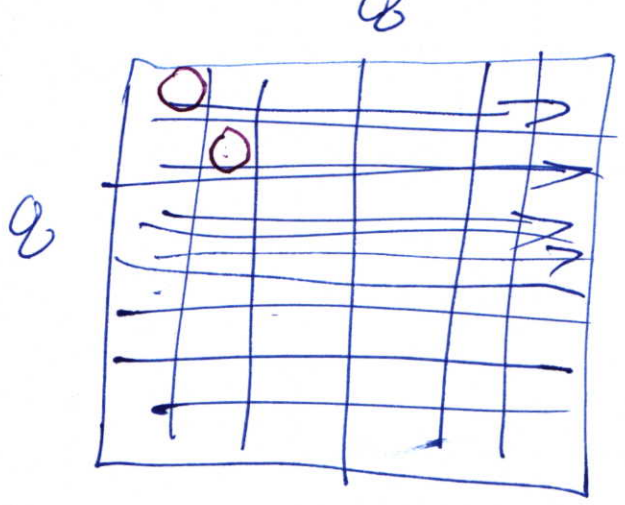


polygon to polygon



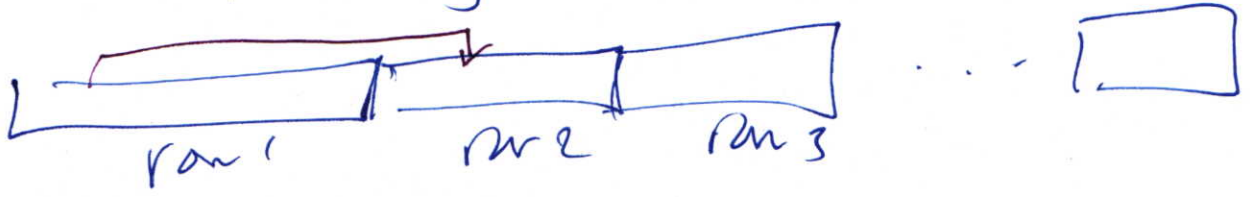
Slide 7

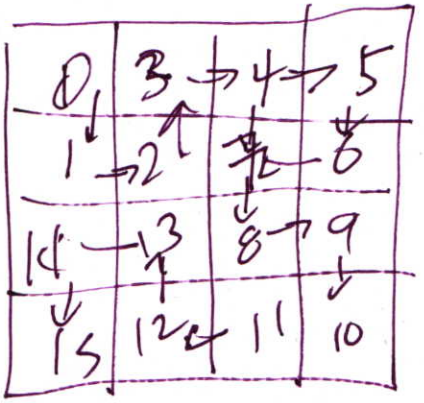
- grid
- qTree
- kd tree



64 boxes

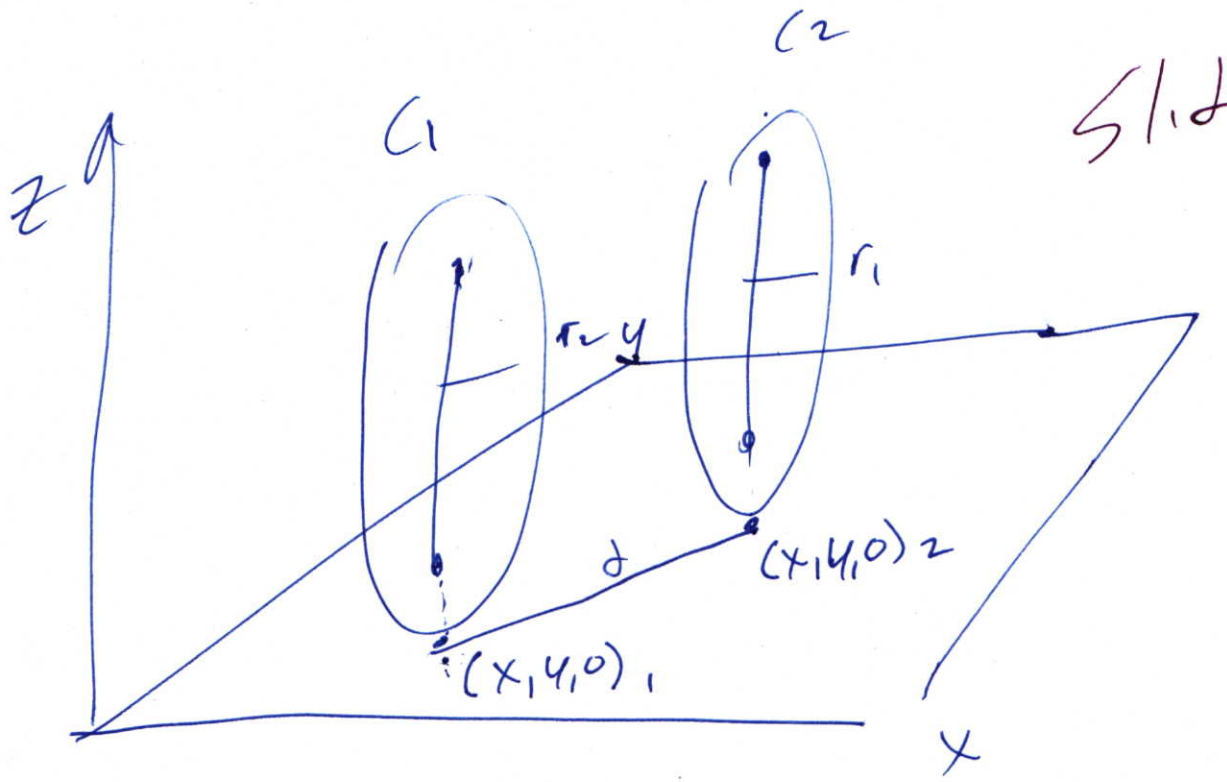
row major order

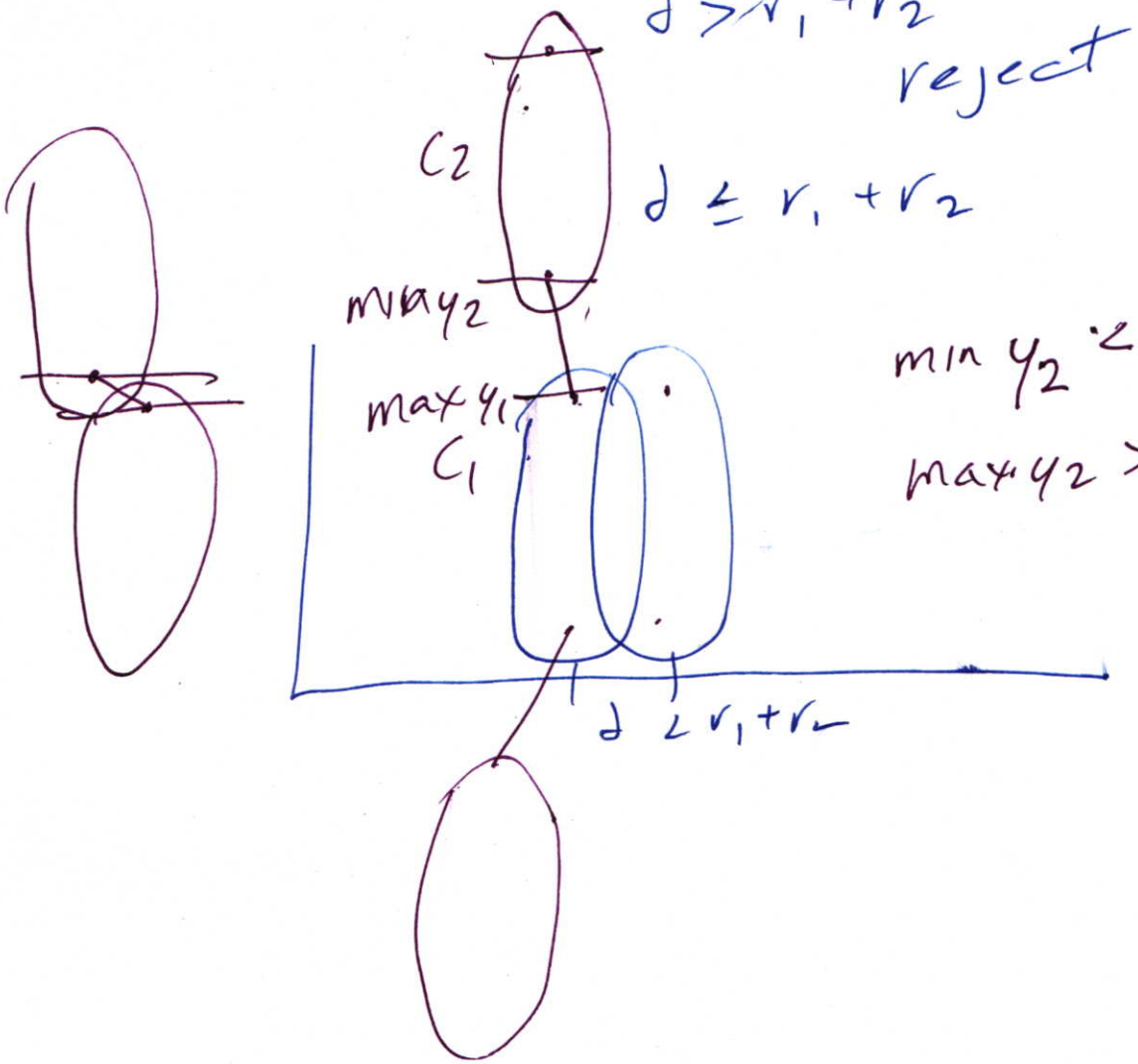




0 1 2 3 4 5 6 7 . .

Slide


C2
C1
z
r1
r2
y
d
(x,y,0)2
(x,y,0)1
x

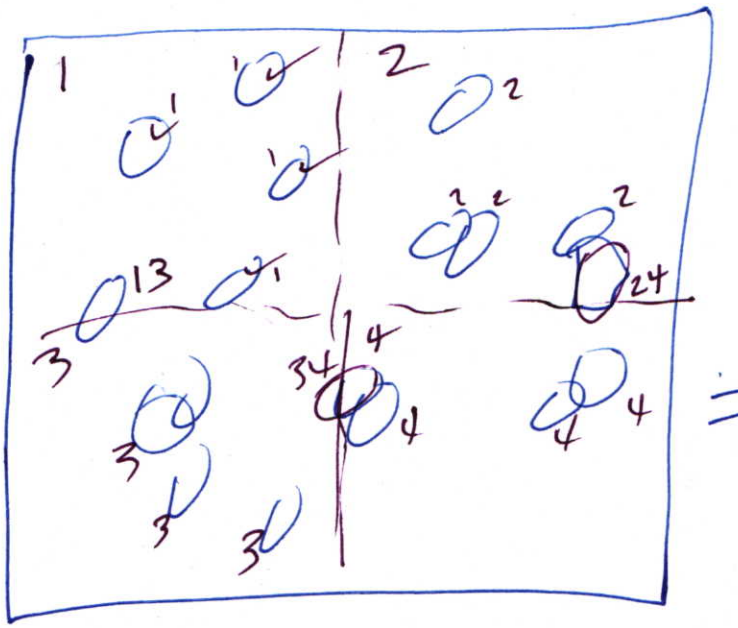

d > r1 + r2
reject
C2
d ≤ r1 + r2
min y2
max y1
min y2 ≤ max y1
max y2 > min y1
C1
d ≤ r1 + r2

Shape x Shape

box vs box

Sphere vs sphere

capsule vs capsule



$O(n^2)$

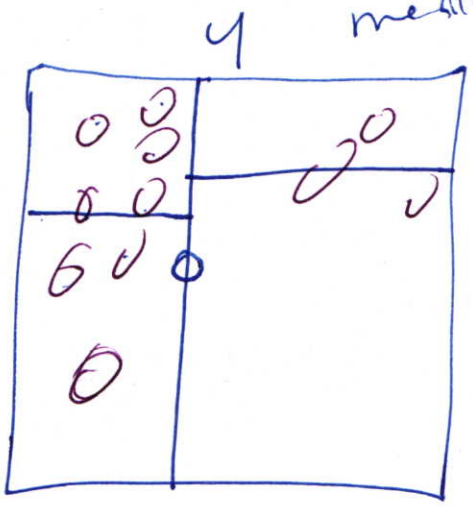
$\Rightarrow \frac{n(n+1)}{2}$

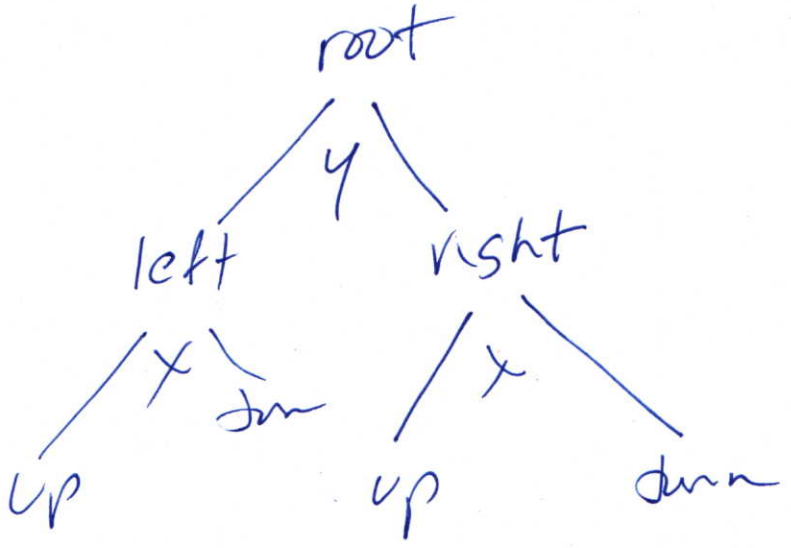
~~[illegible]~~

orginal one shape x n-1 shapes

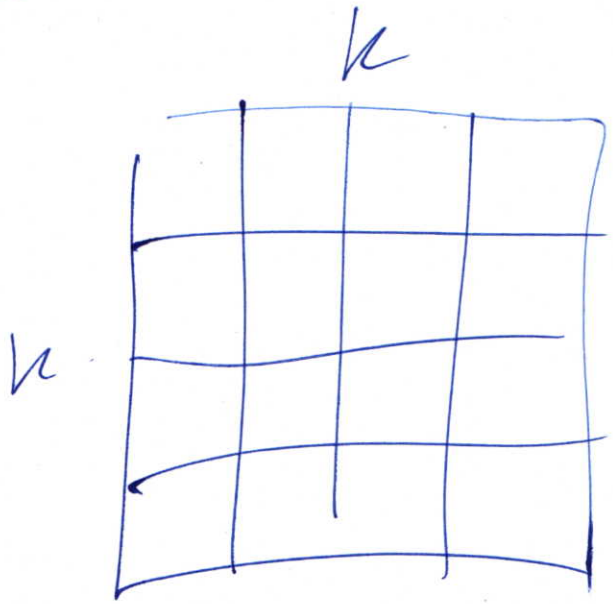
grid
if quadrants one shape x $\frac{n-1}{4}$

k-d tree





Idea 1 - fixed grid



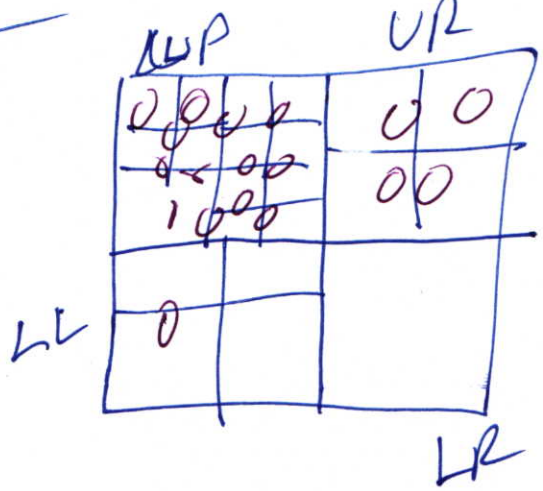
k^2 buckets

one shape $\times \frac{n}{k^2}$

10,000

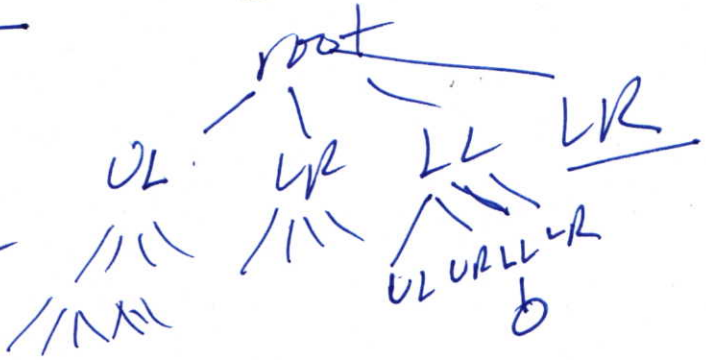
$k = 100$ $k^2 = 10,000$

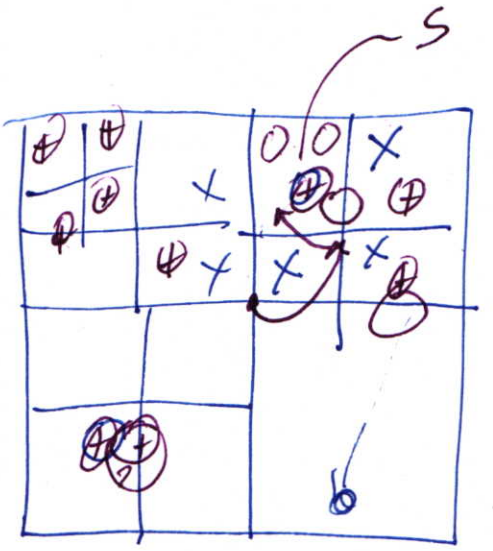
Idea 2 -



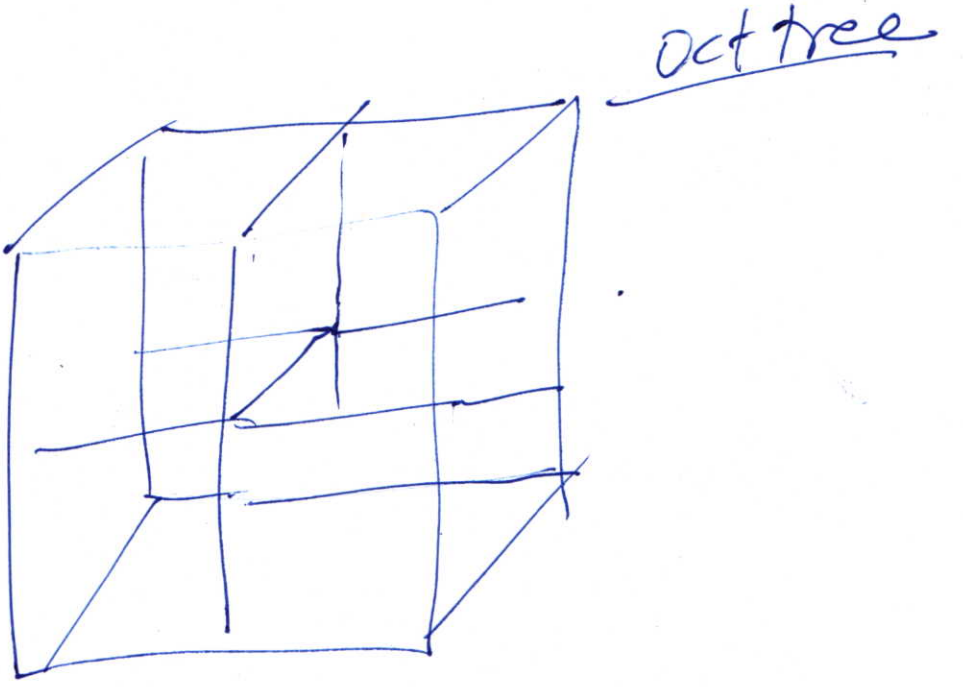
quad trees

k-d trees



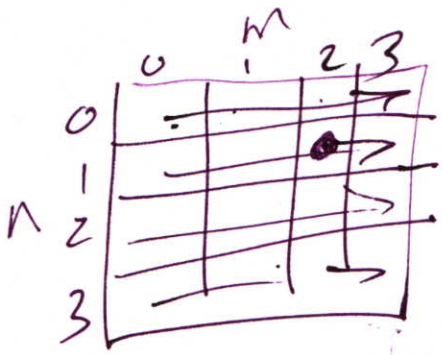


S QT

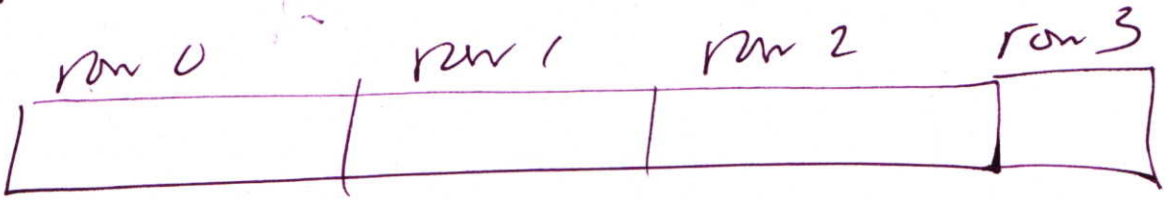


Array mapping function

Amf(i, j) → linear order of an element



Amf(i,j) =



Amf(0,0) = 0

Amf(0,2) = 2

AMF(1,2) = i * m + j

Morton order

AMF(i, j) =

AMF(2, 3) = 010, 011