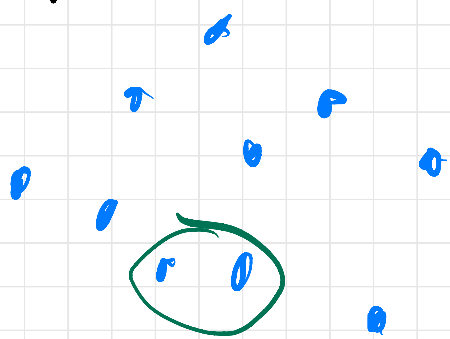
# Closest Pair (in the plane)

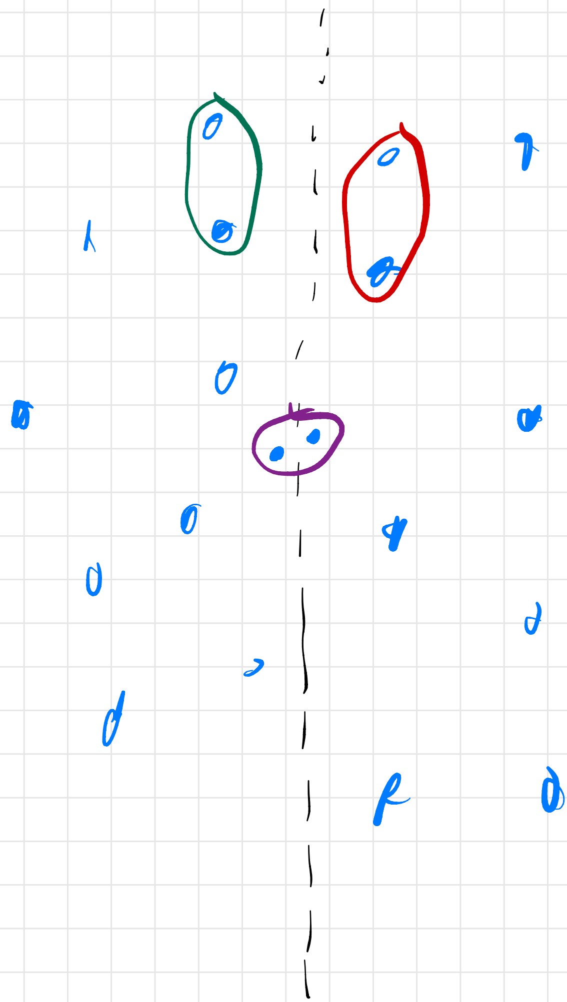
Given $n$ points in the plane as two arrays $X[1..n]$
$Y[1..n]$

ith point at $(X[i], Y[i])$



Goal: find closest pair
(distance only today)

Could check all pairs, but $\Theta(n^2)$ time!



Partition using vertical line through median.

```
ClosestPair(X[1 .. n], Y[1 .. n]):
  if n ≤ 3
      solve by brute force

  XL[1 .. ⌊n/2⌋] and YL[1 .. ⌊n/2⌋] ← leftmost ⌊n/2⌋ points
  ℓ ← ClosestPair(XL[1 .. ⌊n/2⌋], YL[1 .. ⌊n/2⌋]) ⟨⟨Recurse!⟩⟩
  XR[1 .. ⌈n/2⌉] and YR[1 .. ⌈n/2⌉] ← rightmost ⌈n/2⌉ points
  r ← ClosestPair(XR[1 .. ⌈n/2⌉], YR[1 .. ⌈n/2⌉]) ⟨⟨Recurse!⟩⟩

  m ← ∞ ⟨⟨Find closest pair between two halves⟩⟩
  for i ← 1 to ⌊n/2⌋
      for j ← 1 to ⌈n/2⌉
          if Distance(XL[i], YL[i], XR[j], YR[j]) < m
              m ← Distance(XL[i], YL[i], XR[j], YR[j])

  return min {ℓ, r, m}
```

$\Theta(n^2)$

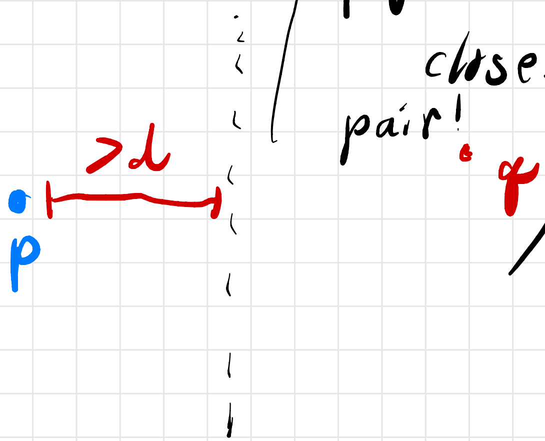
$$T(n) = 2T(n/2) + \Theta(n^2)$$

$$= \Theta(n^2)$$

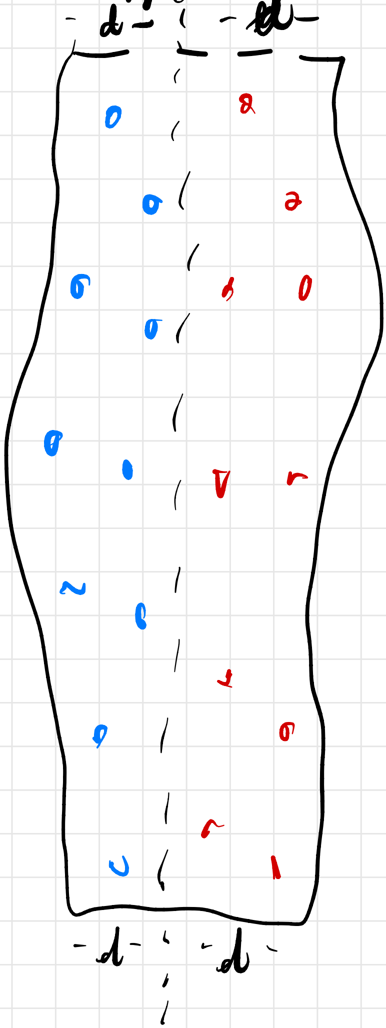
If closest pair has a point on both sides, they are $\leq d$ apart

$d = \min\{\ell, r\}$

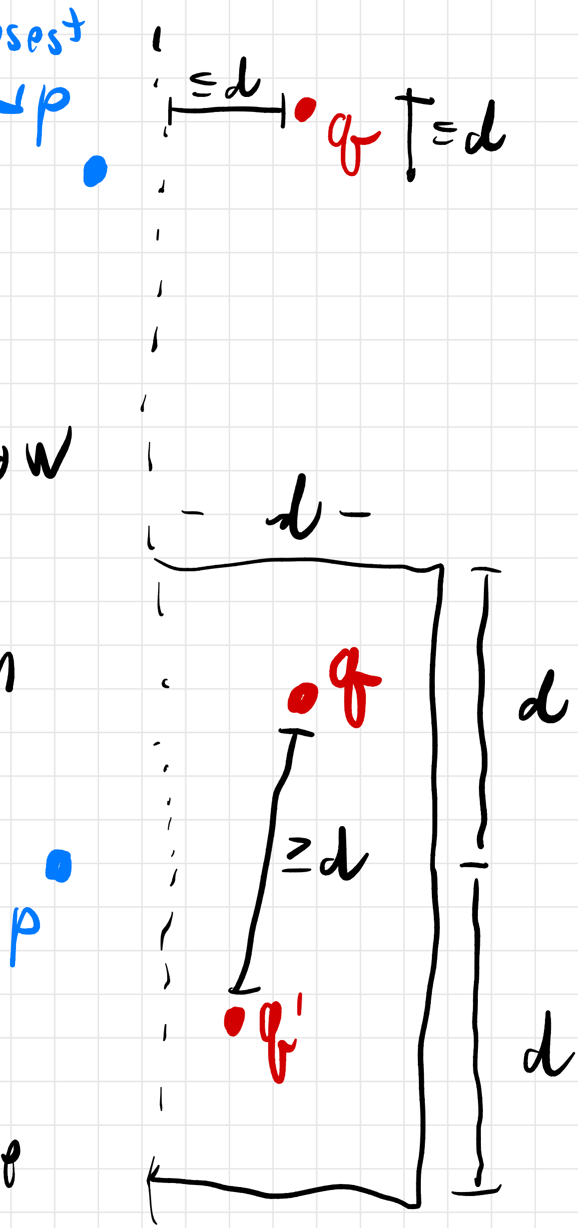
pq can't be closest pair!



So only check pairs with points distance $\leq d$ from vertical line!




But maybe all points are that close!




other member $\leq d$ units above or below p!

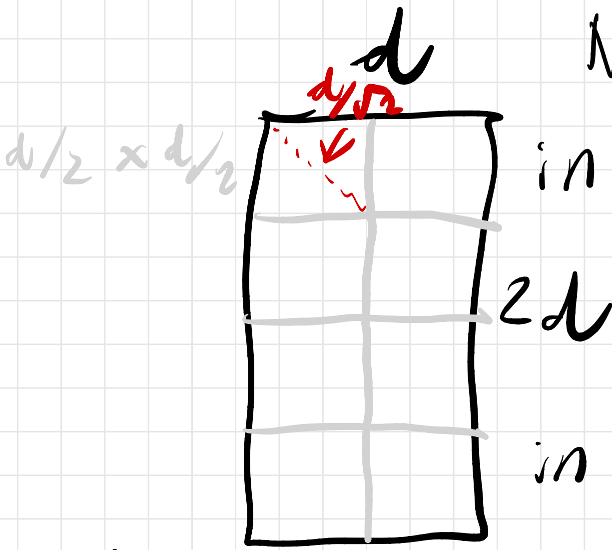
$\Rightarrow$ it lies in a $d \times 2d$ rectangle.

left side on median vertical line

vertically aligned with p at center

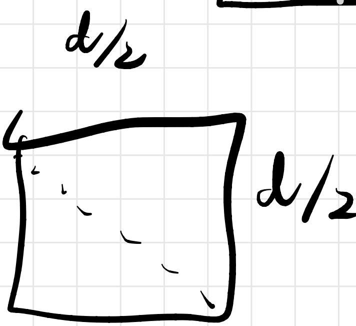
A $d \times 2d$ rect holds $\leq 6$ points of pairwise distance $\geq d$. (packing argument)

Proof of $\leq 8$:




No two points in a $d/2 \times d/2$ square.

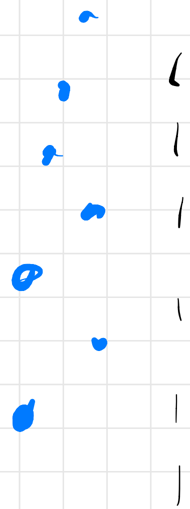
So $\leq 8$ points in $2d \times d$ rect.




$$(d/2) \cdot \sqrt{2} = d/\sqrt{2}$$

Loop over points p by increasing y-coor.

Keep a finger on lowest q in current box.



So vertically scanning both sides.

Sort ahead of top level call.

```
ClosestPairFast(X[1 .. n], Y[1 .. n]):
  ⟨⟨Assumes points come pre-sorted by y-coordinate⟩⟩
  if n ≤ 3
      solve by brute force

  XL[1 .. ⌊n/2⌋] and YL[1 .. ⌊n/2⌋] ← leftmost ⌊n/2⌋ points
  ℓ ← ClosestPairFast(XL[1 .. ⌊n/2⌋], YL[1 .. ⌊n/2⌋]) ⟨⟨Recurse!⟩⟩
  XR[1 .. ⌈n/2⌉] and YR[1 .. ⌈n/2⌉] ← rightmost ⌈n/2⌉ points
  r ← ClosestPairFast(XR[1 .. ⌈n/2⌉], YR[1 .. ⌈n/2⌉]) ⟨⟨Recurse!⟩⟩
  d ← min {ℓ, r}

  ⟨⟨Find closest pair between two halves⟩⟩
  XL′[1 .. k] and YL′[1 .. k] ← subset of leftmost ⌊n/2⌋ points with x-coordinate ≥ XR[1] − d
  XR′[1 .. o] and YR′[1 .. o] ← subset of rightmost ⌈n/2⌉ points with x-coordinate ≤ XR[1] + d
  m ← ∞
  jmin ← 1
  for i ← 1 to k
      while jmin ≤ o and YR′[jmin] < YL′[i] − d
          jmin ← jmin + 1
      j ← jmin
      while j ≤ o and YR′[j] ≤ YL′[i] + d
          if Distance(XL′[i], YL′[i], XR′[j], YR′[j]) < m
              m ← Distance(XL′[i], YL′[i], XR′[j], YR′[j])
          j ← j+1
  return min {ℓ, r, m}
```

O(n) increments TOTAL

O(8) time

$$T(n) \le 2T(n/2) + O(n)$$

$$\le O(n \log n) = o(n^2)$$

little-oh

# Multiplication

$m$ : a non-neg. integer

Given $x$ & $y$ let $a, b, c, d$ s.t.

$$x = (10^m a + b)$$
$$y = (10^m c + d)$$

$$x \cdot y = 10^{2m} ac + 10^m (bc + ad) + bd$$

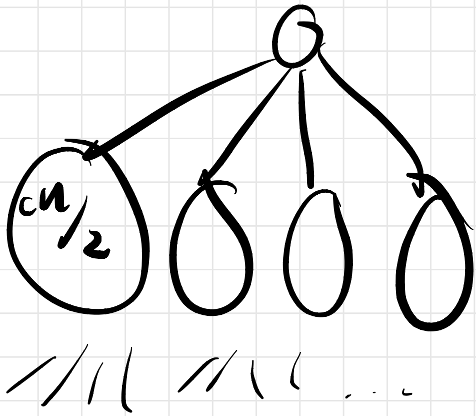
$\underline{\textsc{SplitMultiply}(x, y, n):}$

x & y have $\leq n$ digits

if $n = 1$
    return $x \cdot y$
else
    $m \leftarrow \lceil n/2 \rceil$
    $a \leftarrow \lfloor x/10^m \rfloor$; $b \leftarrow x \bmod 10^m$    $\langle\langle x = 10^m a + b \rangle\rangle$
    $c \leftarrow \lfloor y/10^m \rfloor$; $d \leftarrow y \bmod 10^m$    $\langle\langle y = 10^m c + d \rangle\rangle$
    $e \leftarrow \textsc{SplitMultiply}(a, c, m)$
    $f \leftarrow \textsc{SplitMultiply}(b, d, m)$
    $g \leftarrow \textsc{SplitMultiply}(b, c, m)$
    $h \leftarrow \textsc{SplitMultiply}(a, d, m)$
    return $10^{2m} e + 10^m (g + h) + f$

$$T(n) = 4\,T(n/2) + O(n)$$

$= cn$




$= 2cn$

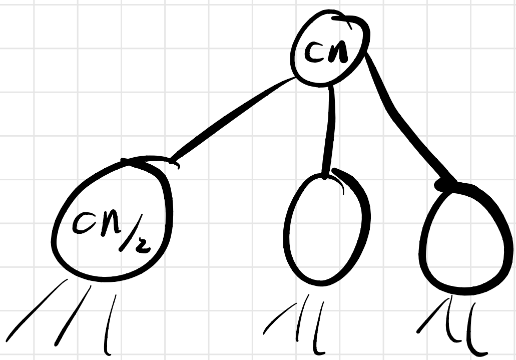
$= 4cn$

$$T(n) = O(n^{\log_2 4}) = O(n^2)$$

$$bc + ad = ac + bd - (a-b)(c-d)$$

```
FASTMULTIPLY(x, y, n):
    if n = 1
        return x · y
    else
        m ← ⌈n/2⌉
        a ← ⌊x/10^m⌋;  b ← x mod 10^m        ⟨⟨x = 10^m a + b⟩⟩
        c ← ⌊y/10^m⌋;  d ← y mod 10^m        ⟨⟨y = 10^m c + d⟩⟩
        e ← FASTMULTIPLY(a, c, m)
        f ← FASTMULTIPLY(b, d, m)
        g ← FASTMULTIPLY(a − b, c − d, m)
        return 10^{2m} e + 10^m (e + f − g) + f
```

$$T(n) = 3T(n/2) + O(n)$$




$cn$

$\frac{3cn}{2}$

$\left(\frac{9}{4}cn\right)$

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

$$\# \text{ leaves} = 3^{\log_2 n} = n^{\log_2 3} = o(n^2)$$

Harvey + van der Hoeven ['20]

: $O(n \log n)$