L6

4102

sep 12 2013
shelat

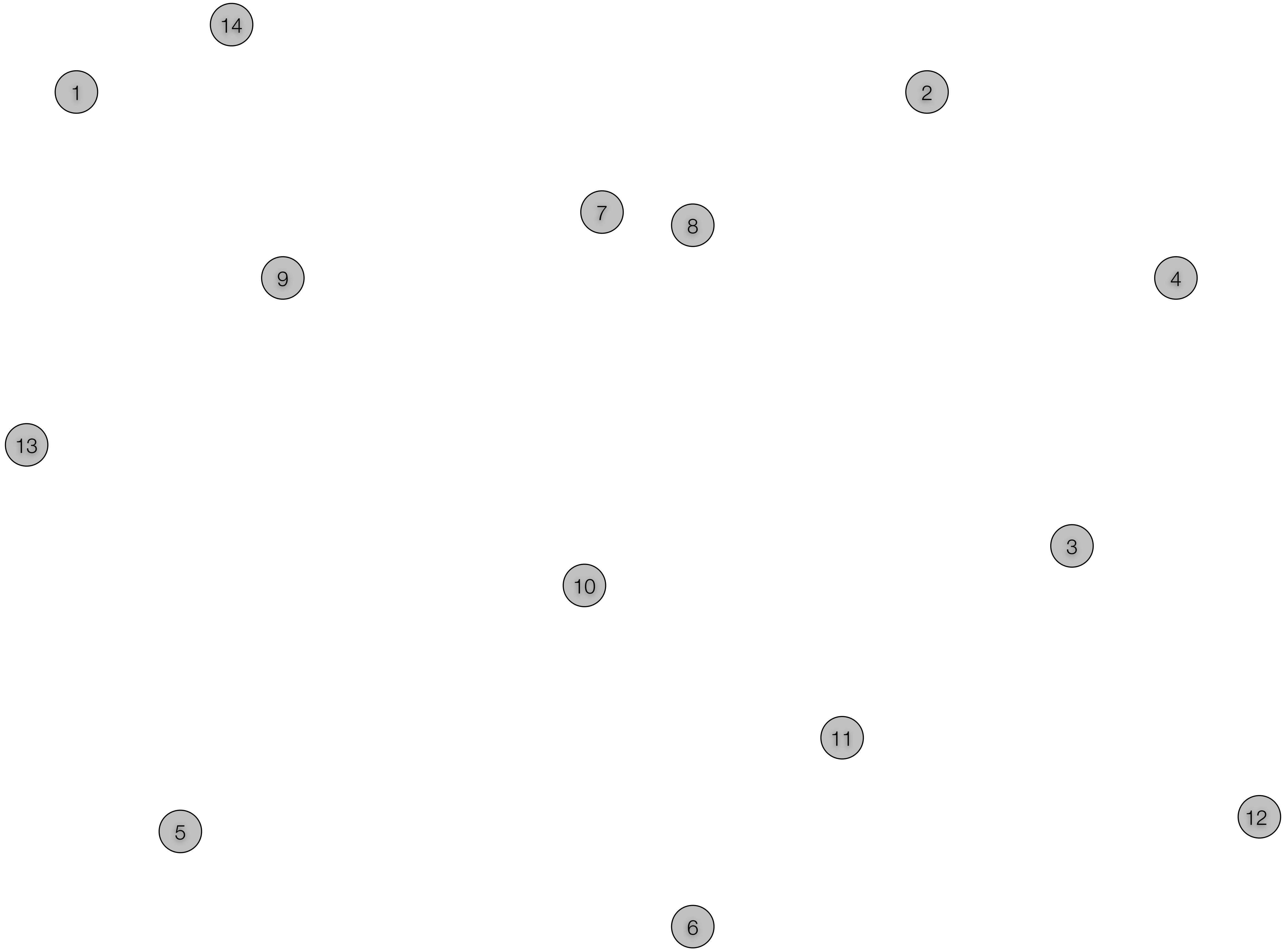
hello

divide&conquer
closest points
matrixmult
median

closest pair

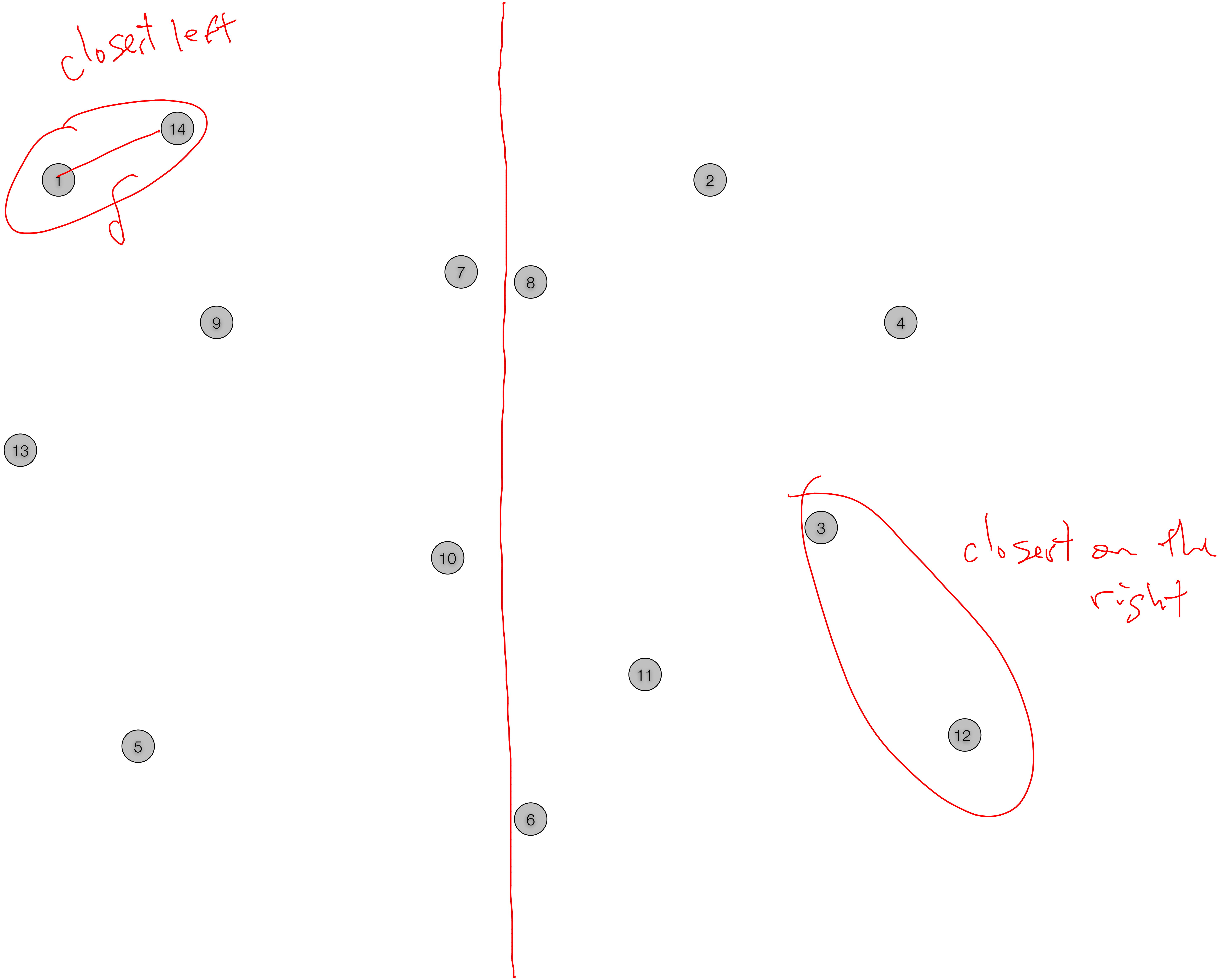
of points


simple brute force approach takes $\Theta(n^2)$

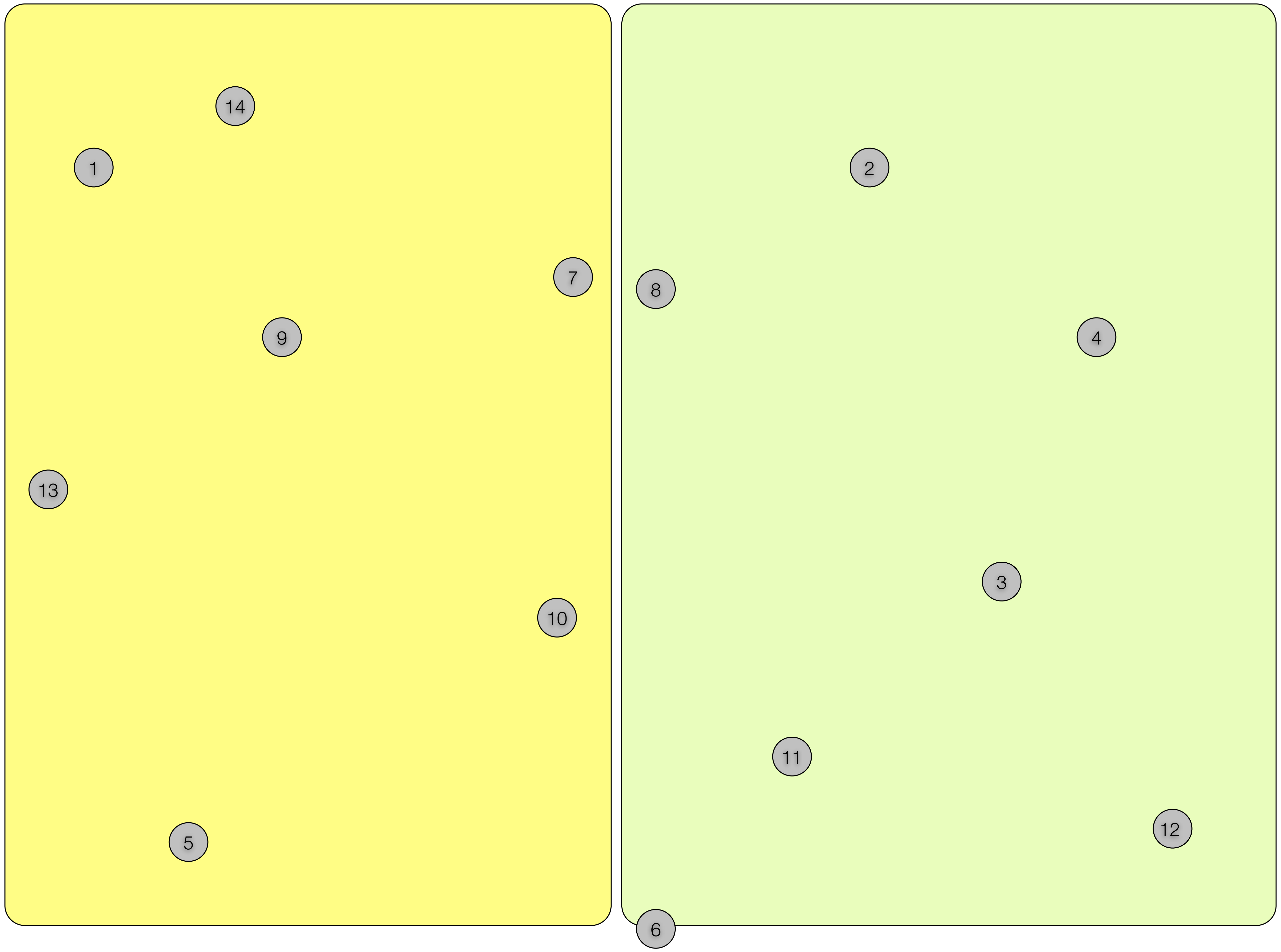


solve the large problem by

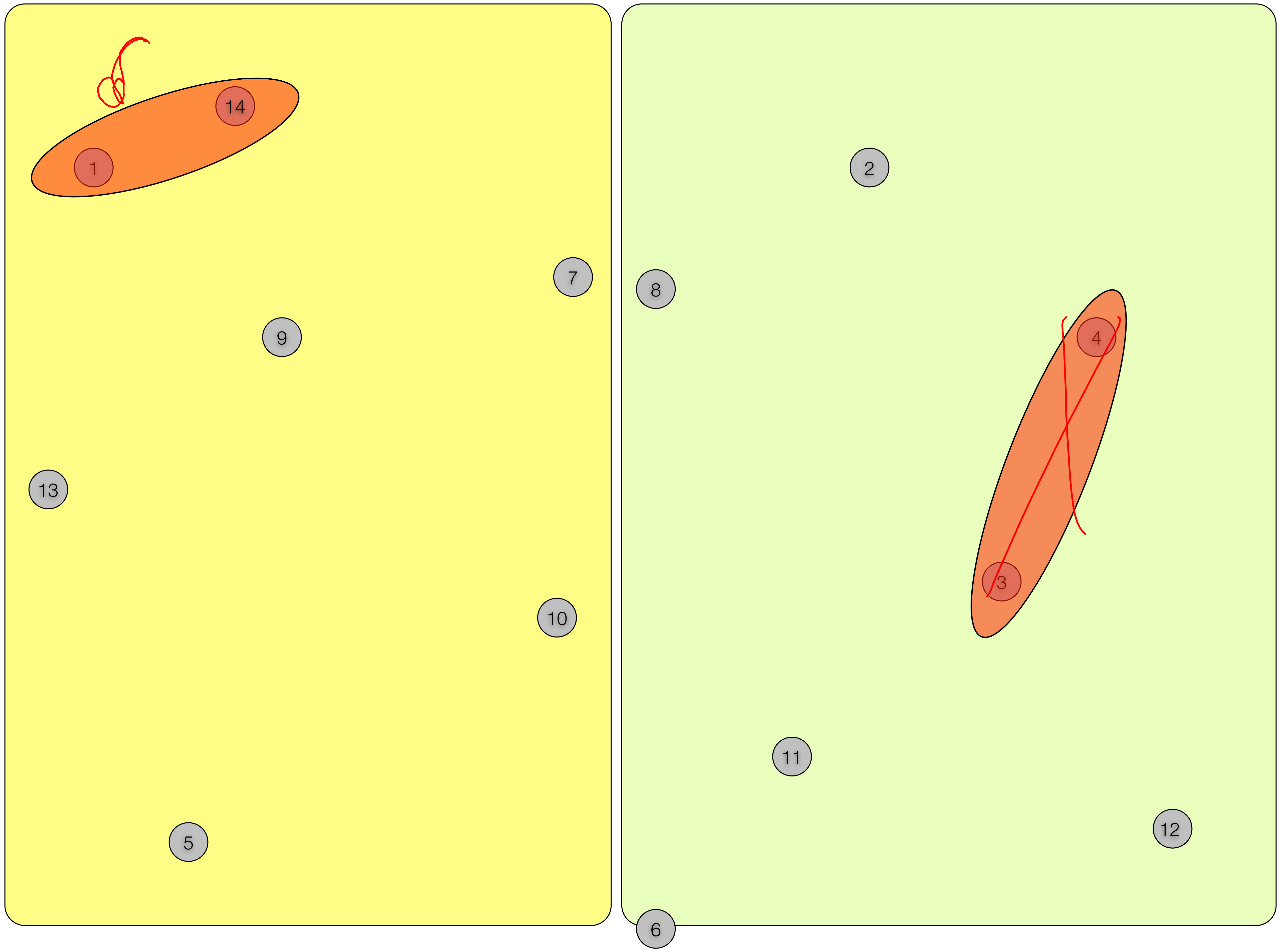
solving smaller problems
and combining solutions


closest left
14
1
2
7
8
9
4
13
3
closest on the right
10
11
12
5
6

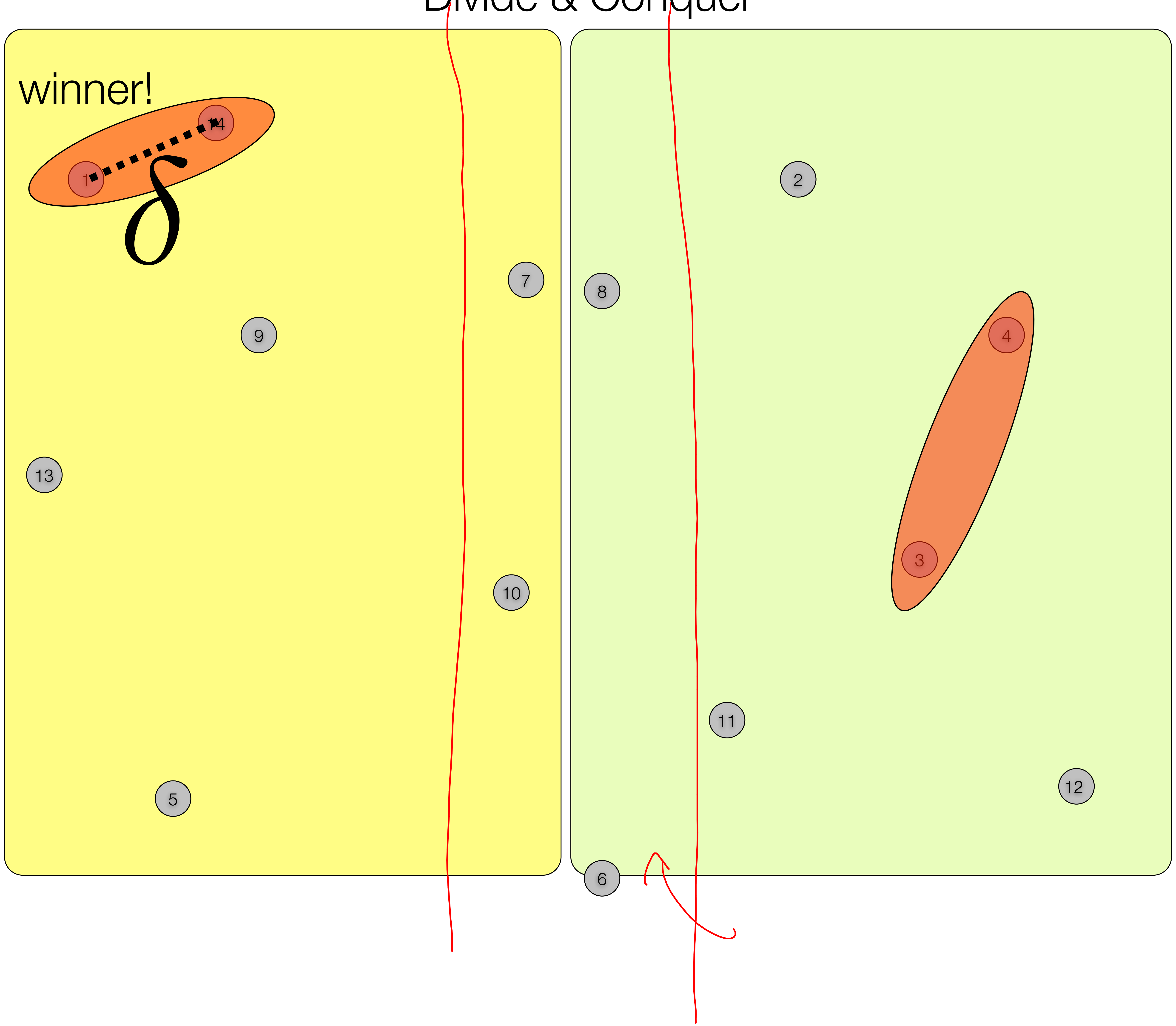
Divide & Conquer

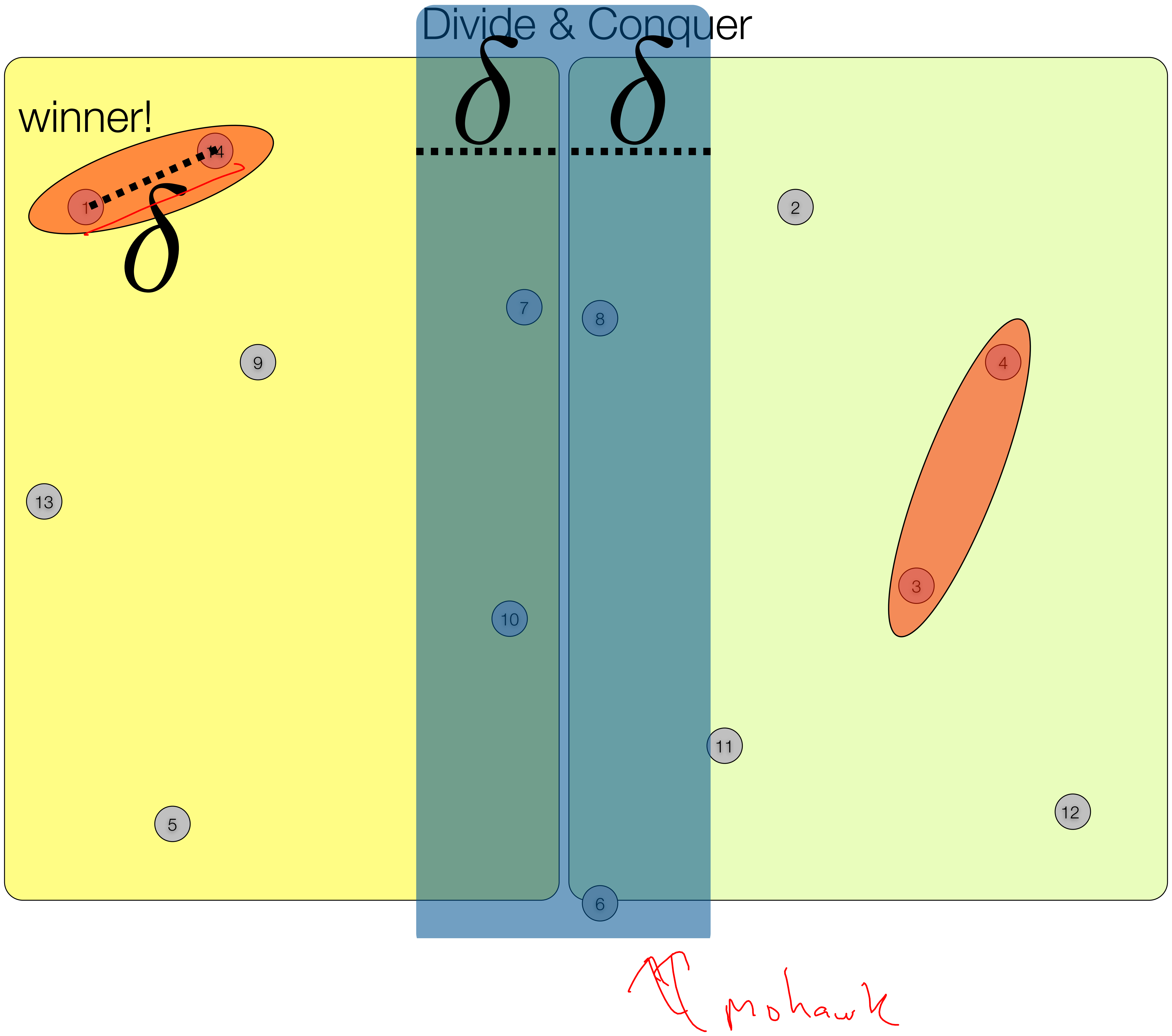


Divide & Conquer

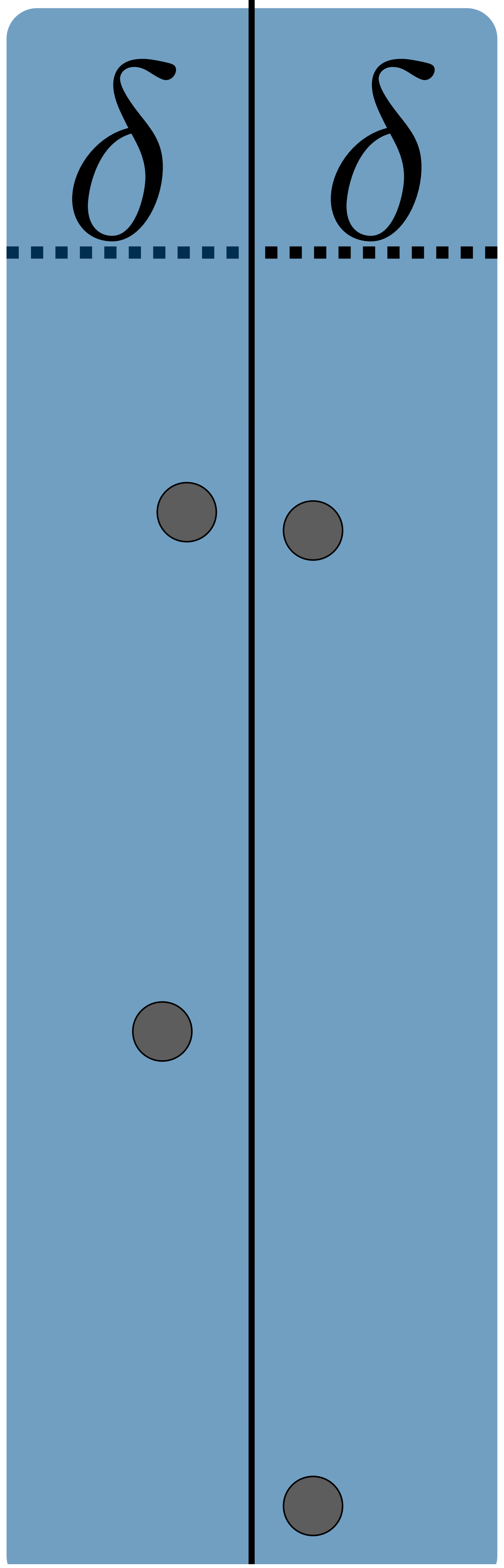


Divide & Conquer

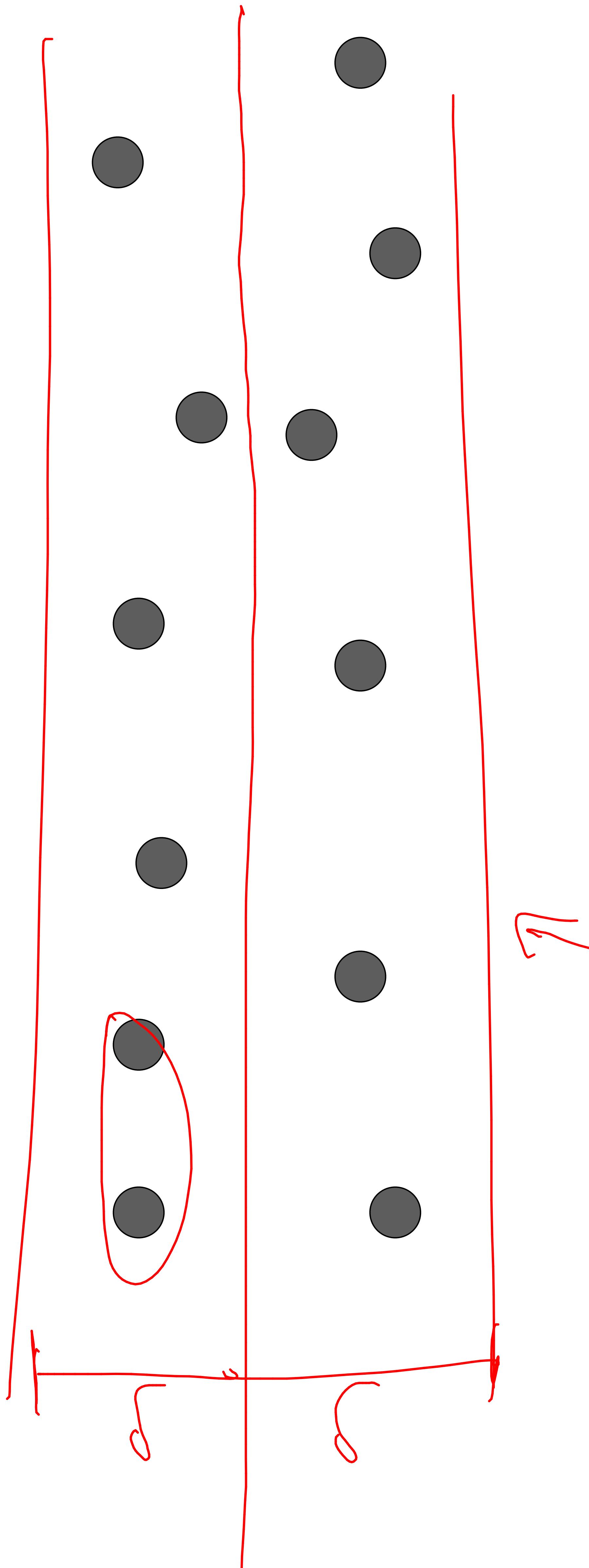




Divide & Conquer
winner!
δ
δ
δ
1
14
2
7
8
9
4
13
3
10
11
5
12
6
mohawk



exhaustively trying each pair $\Rightarrow \Theta(n^2)$

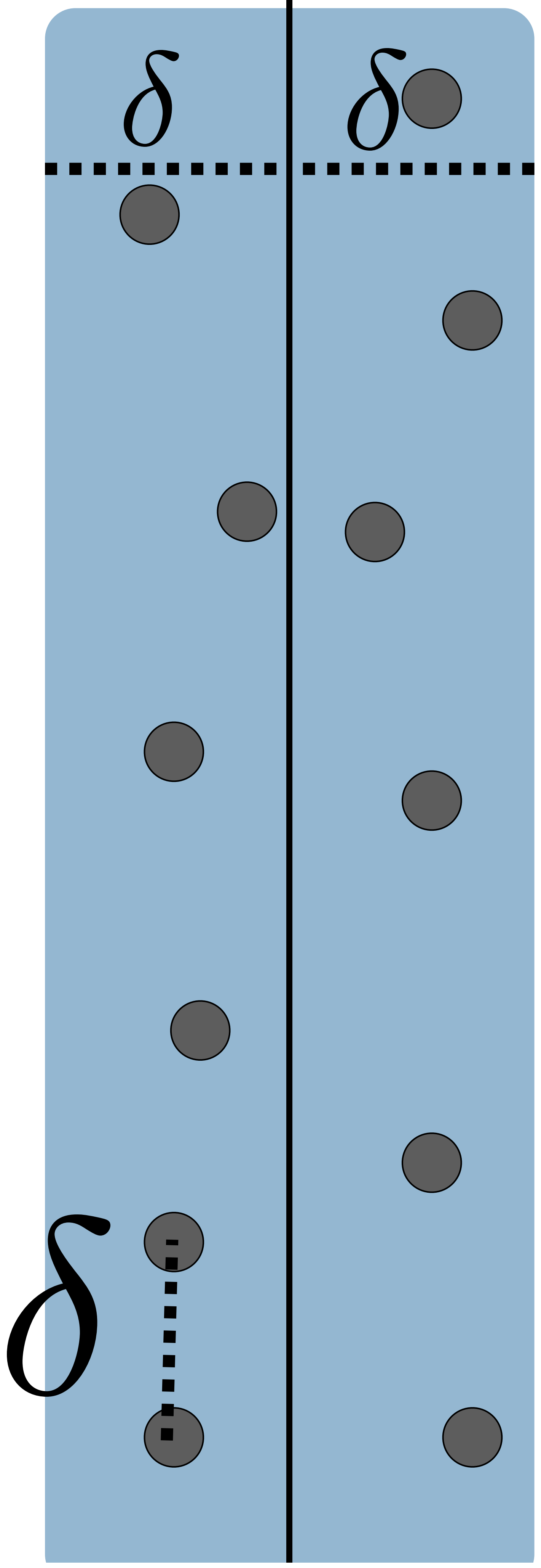


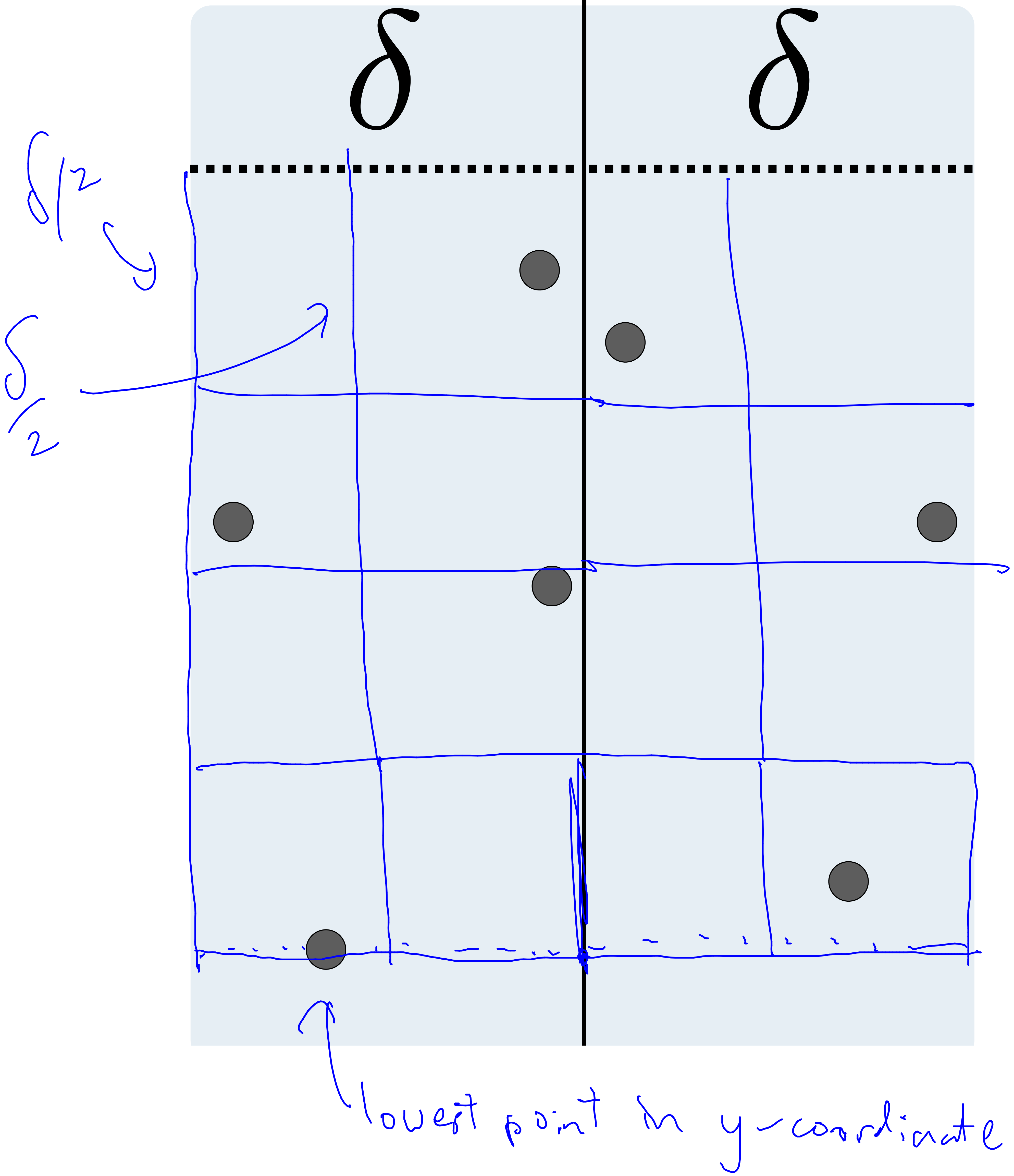
We need: is a linear time approach to detecting a Romeo-Juliet winner

$\Theta(n)$ time

inspecting these points

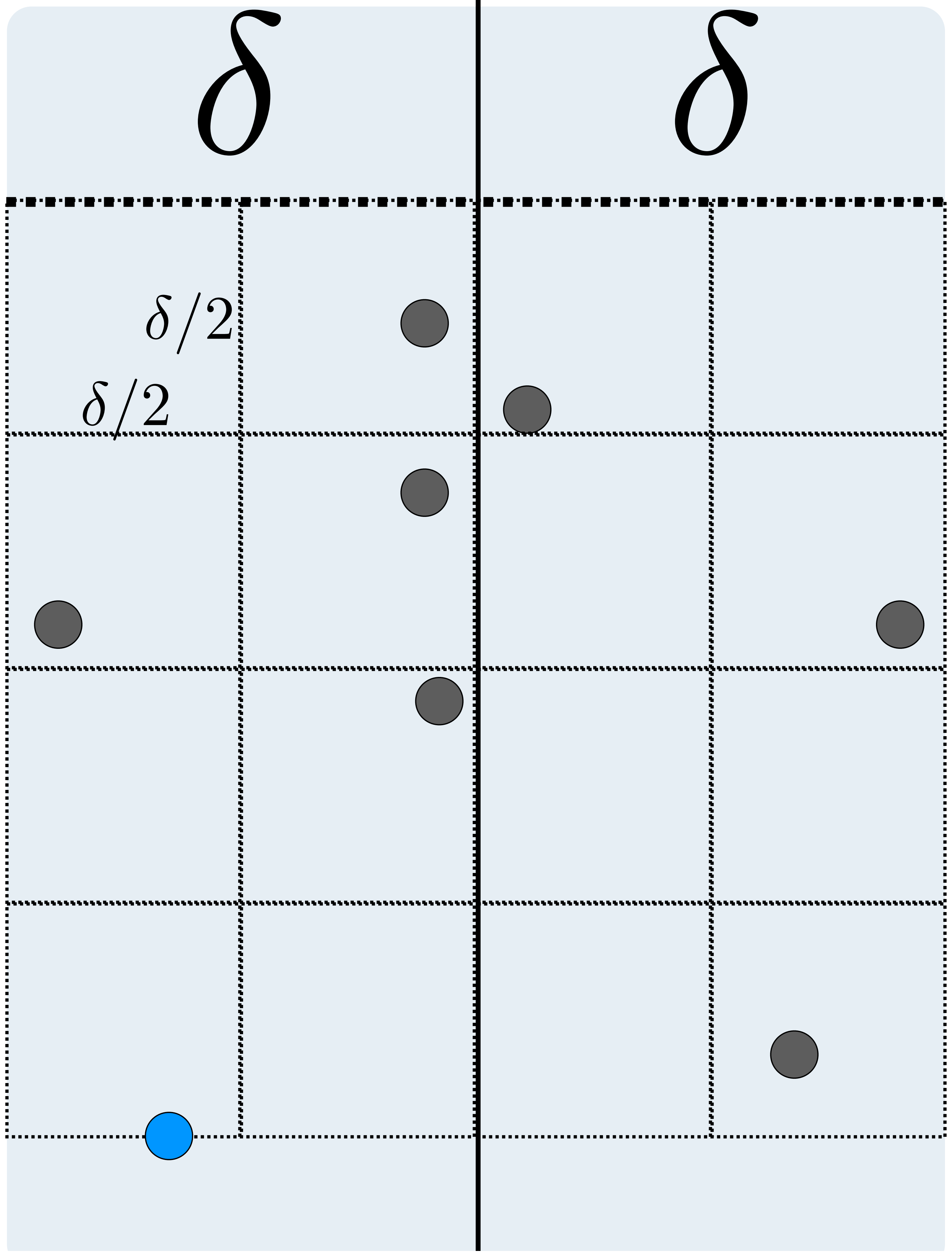
We know the answer $< \delta$

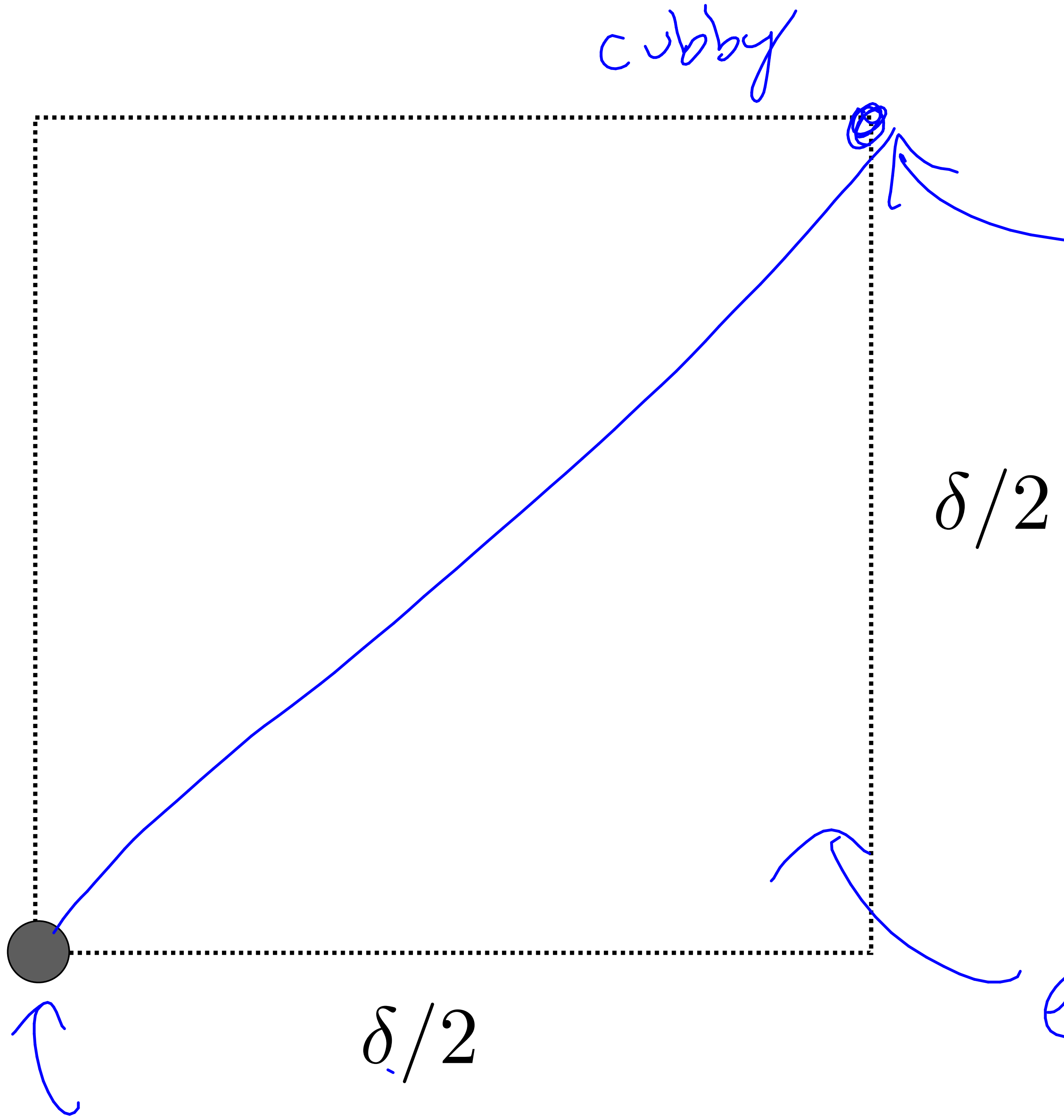

δ
δ
δ



Imaginary grid for our reasoning.

Imagine there is a grid of cubbies starting at the lowest Y point



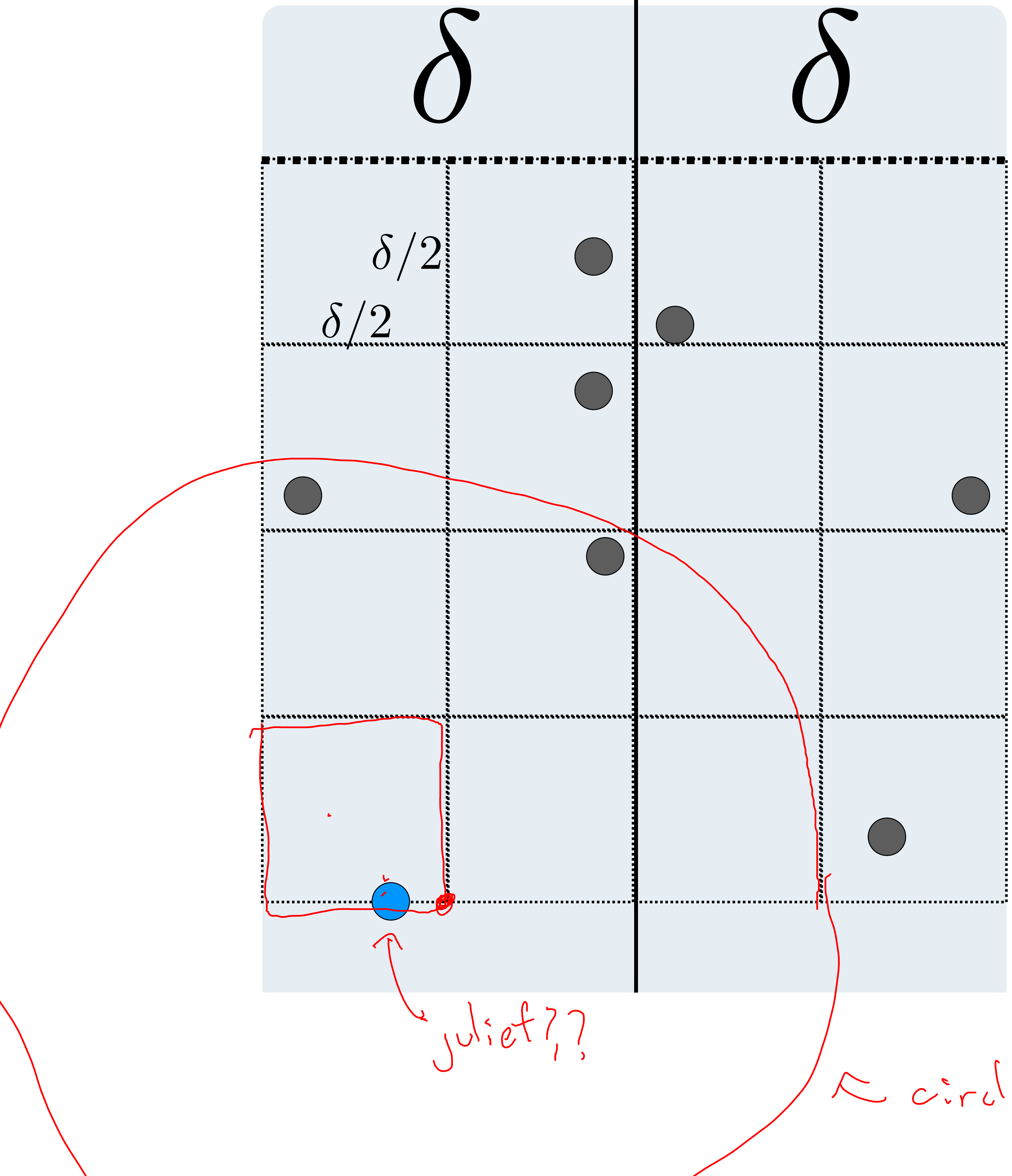


this point will be

$$\sqrt{\left(\frac{\delta}{2}\right)^2 + \left(\frac{\delta}{2}\right)^2} = \frac{\sqrt{2}}{2} \cdot \delta < \underline{\delta}$$

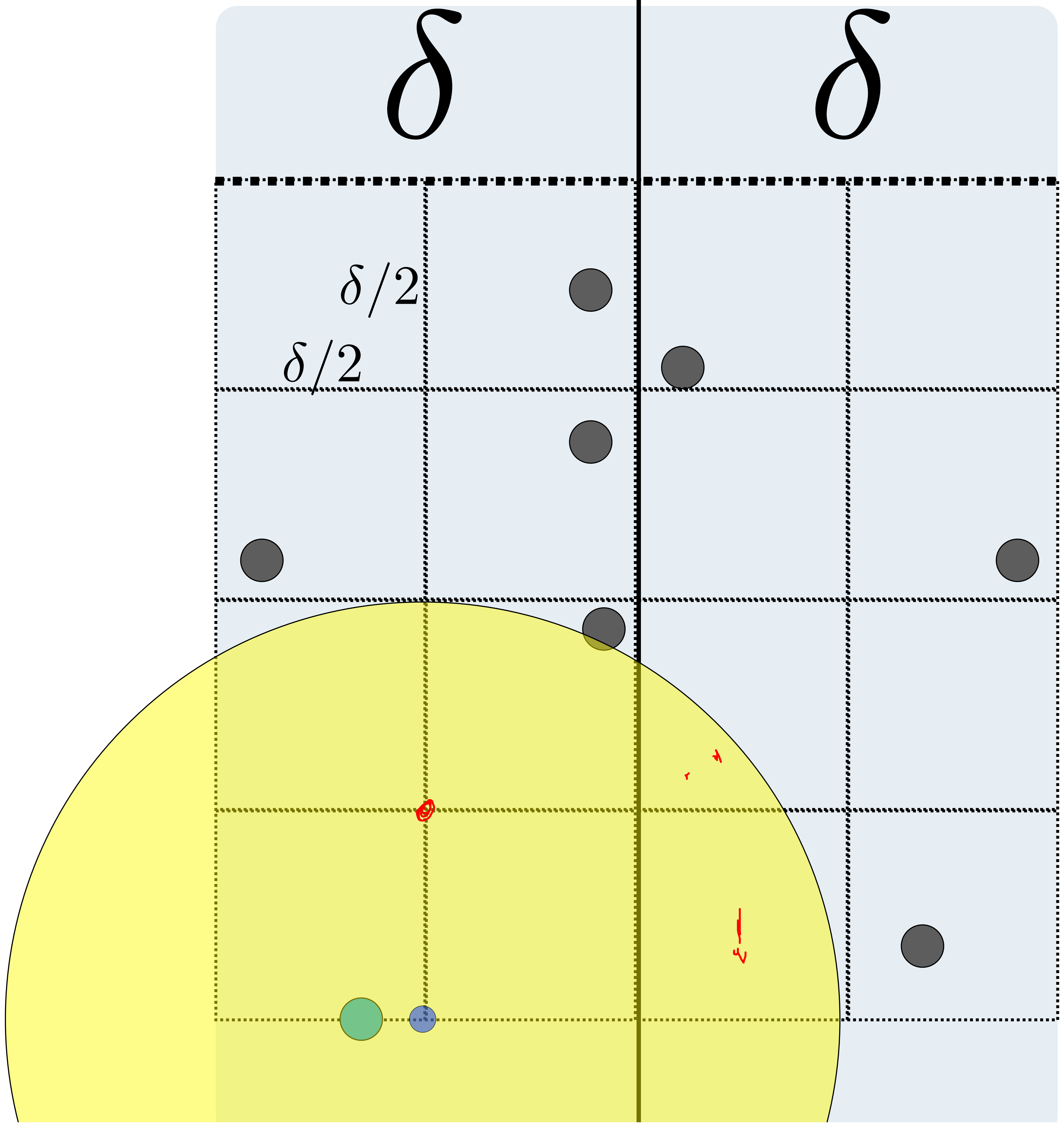
each cubby can have only 1 point in it (at most)

Lonely cubby property.

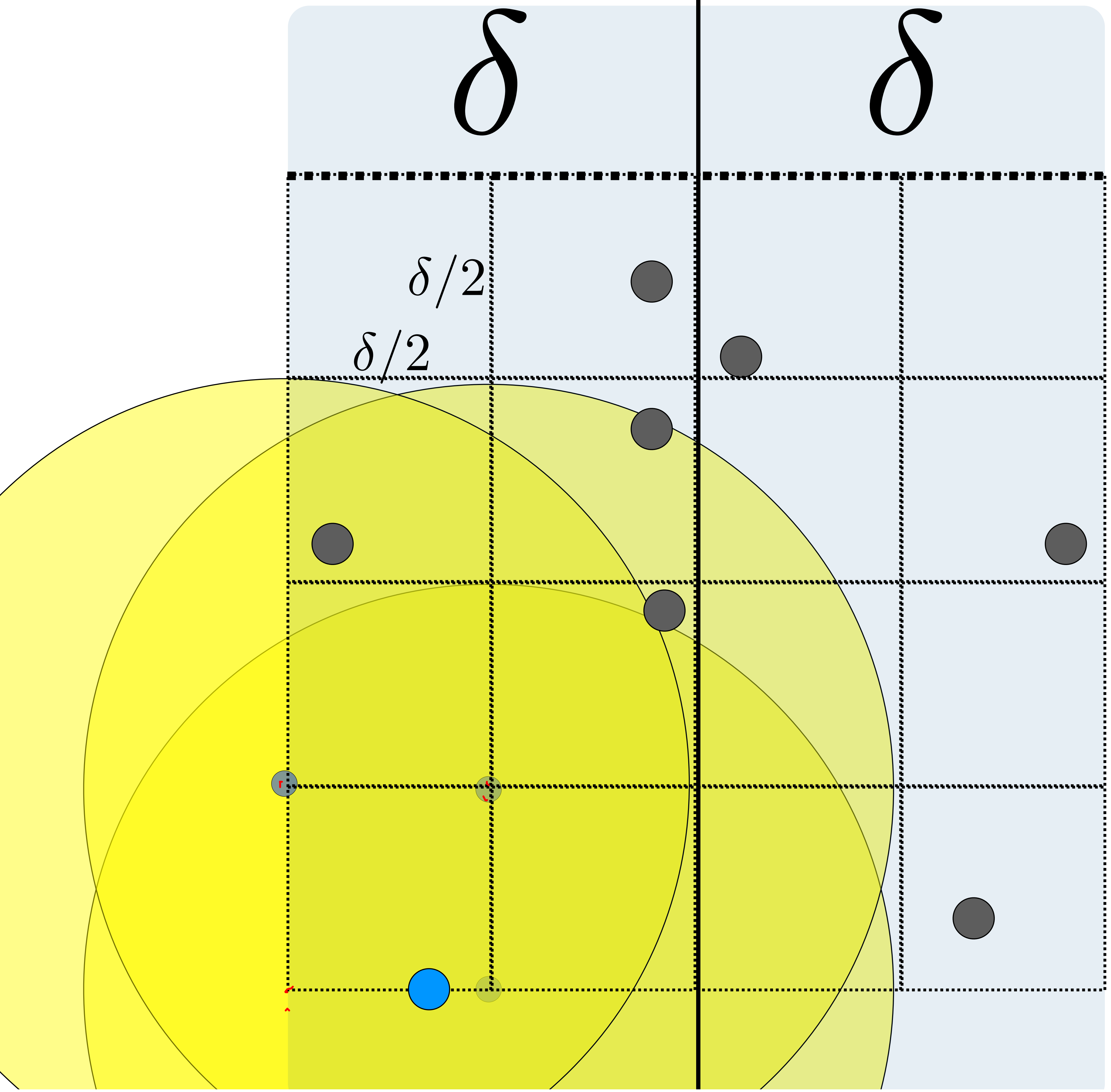


FACT: At most 1 point in each cubby

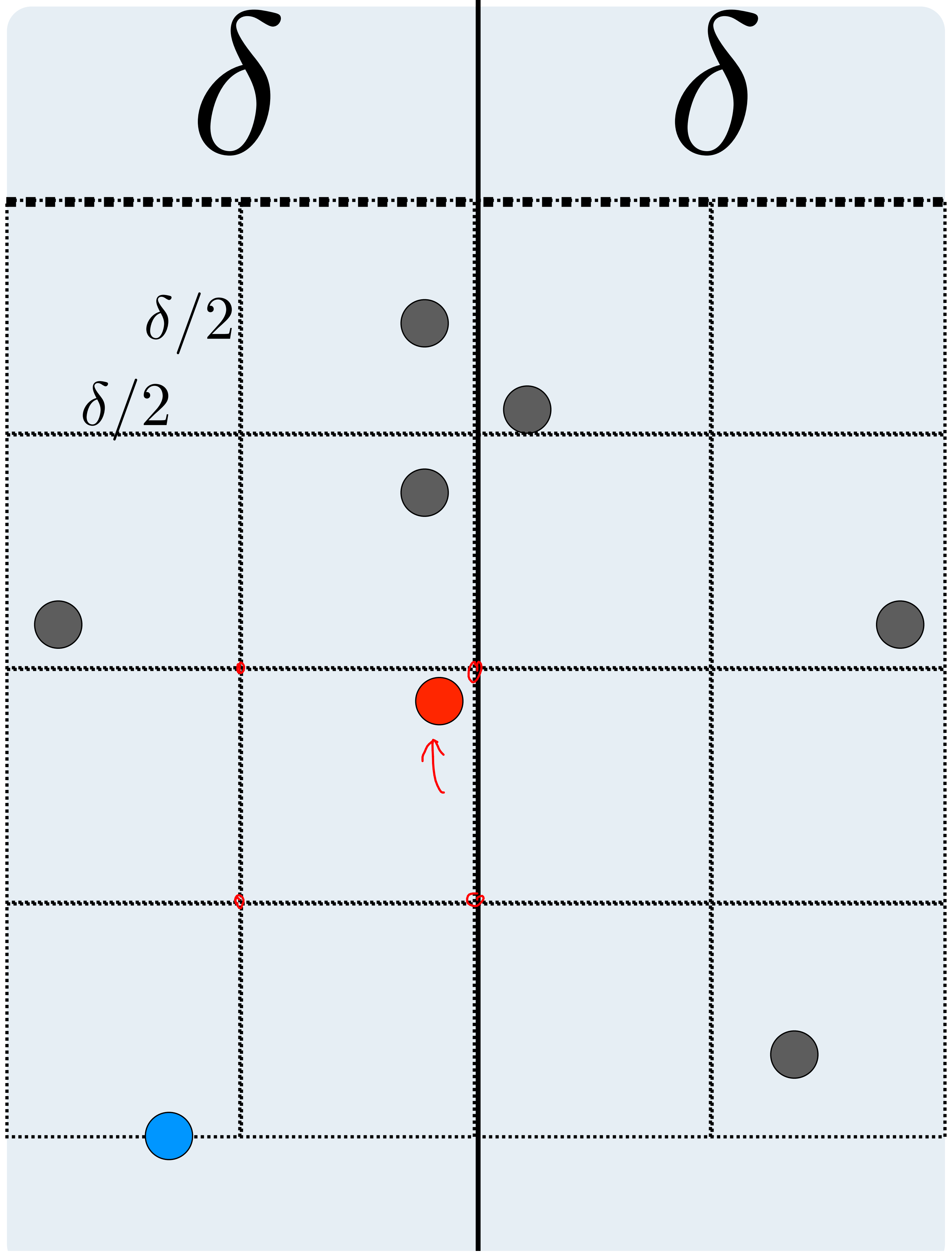
← circle of radius δ centered at the corner



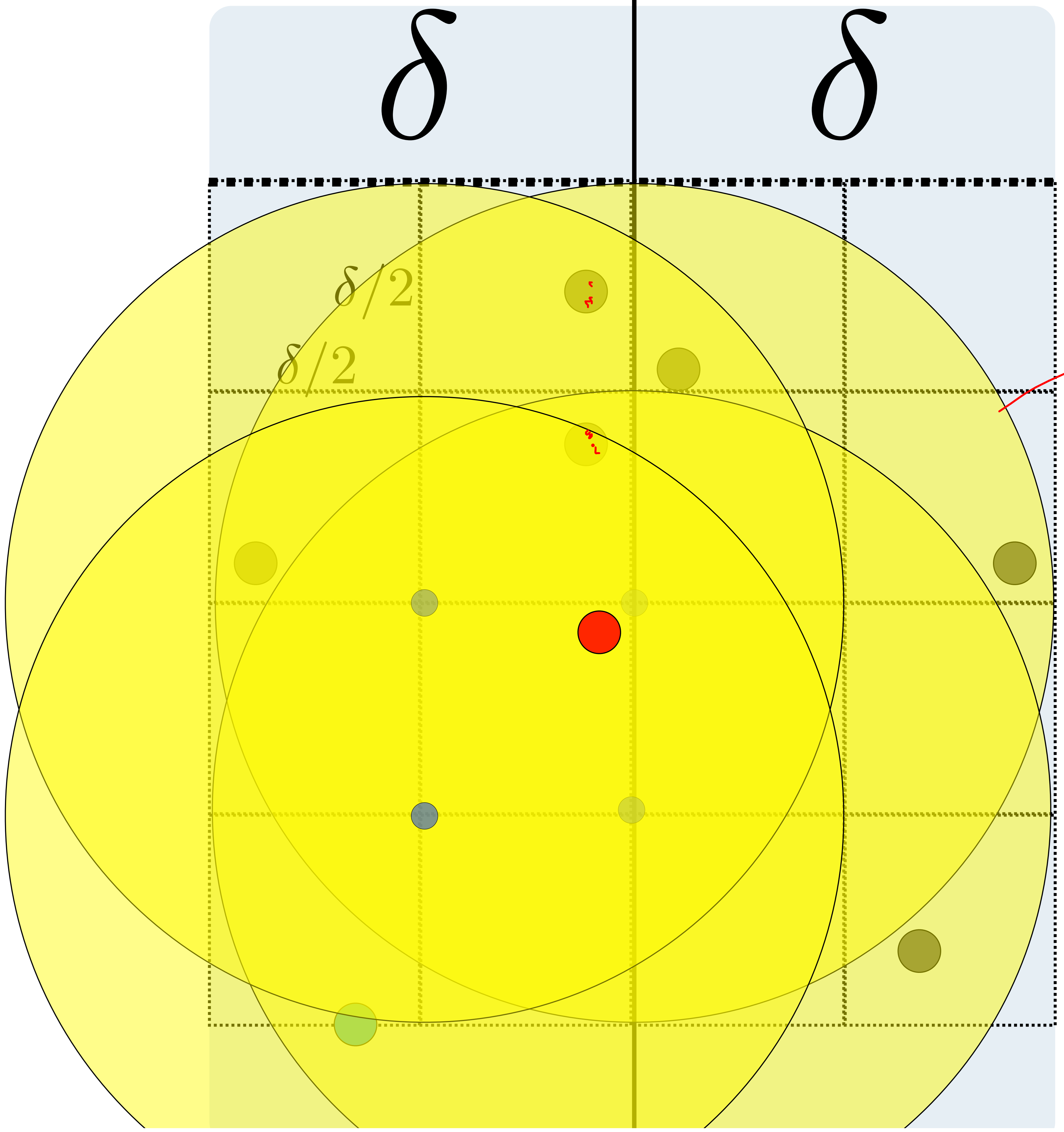
FACT: <=1 point per cubby



FACT: <=1 point per cubby



δ
δ
$\delta/2$
$\delta/2$



→ radius δ

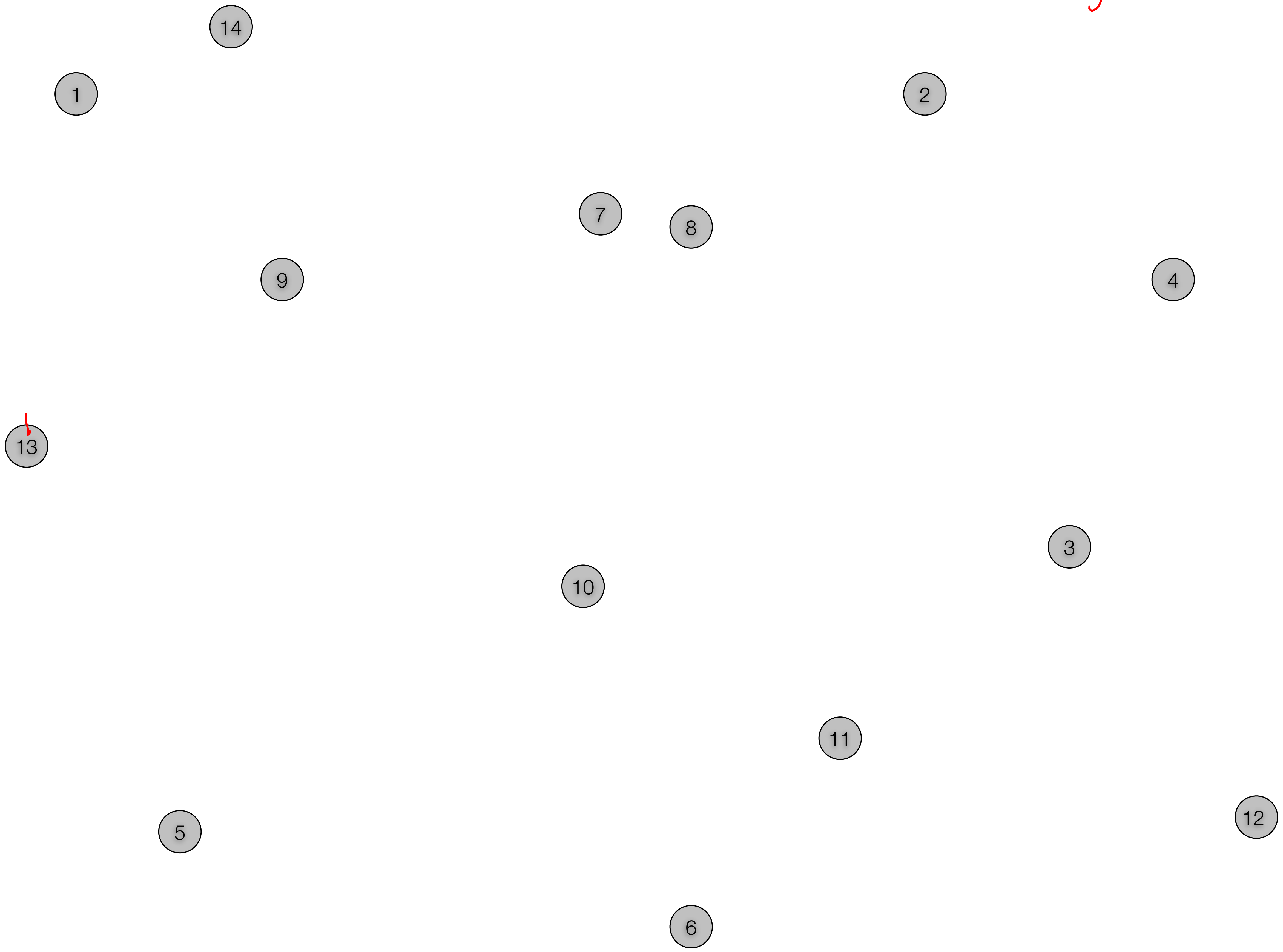
16 squares (one of which is mrs. red's own square) larger.

⇒ there are only 15 possible cubbies for mrs. red's candidate Romeo

Details: How to do step 1?

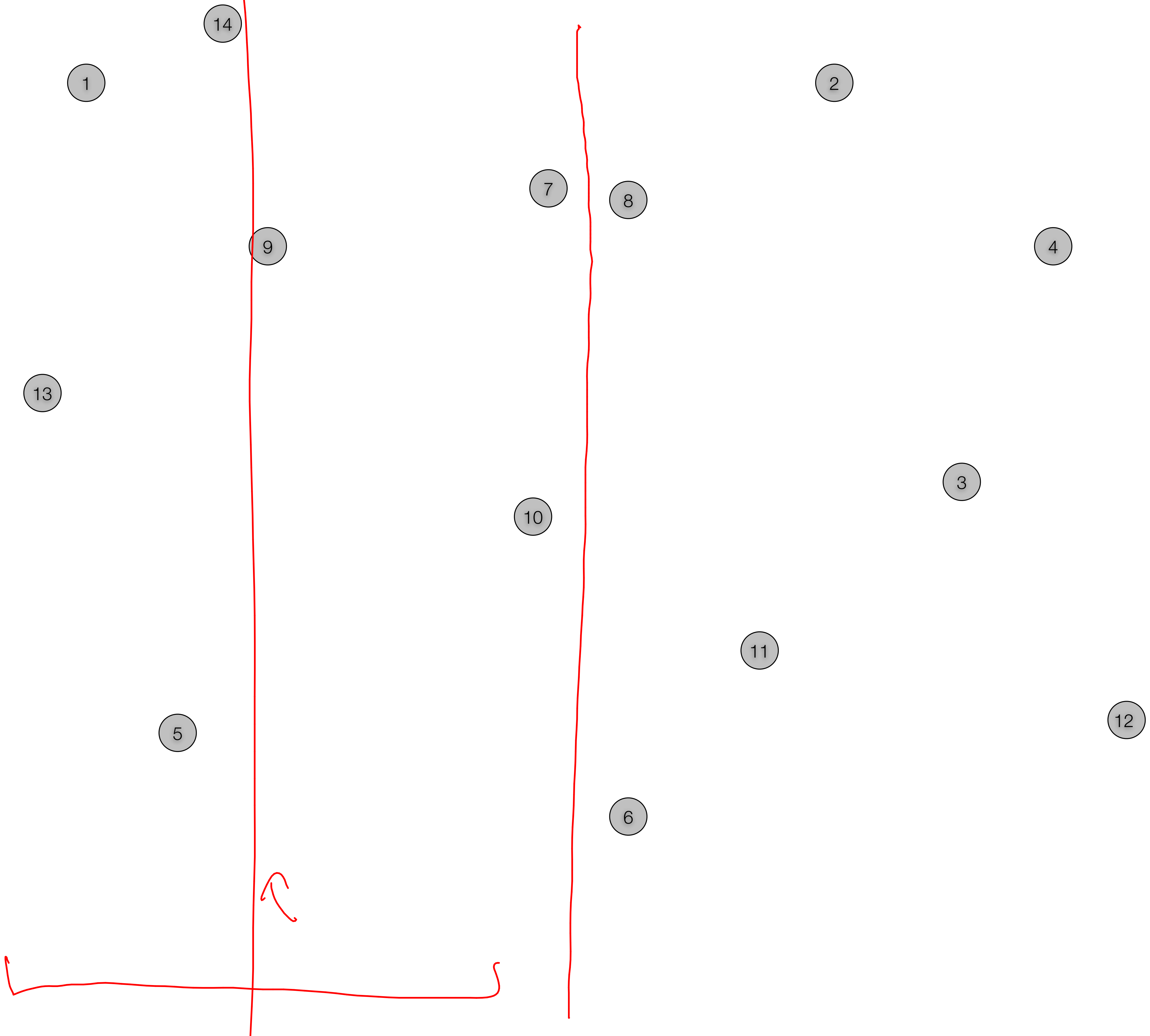
MergeSort point in X coord: 13 1 5 14 9 10 7 6 8 11 2 3 4 12

sort in y coord:



sorted in X: 13 1 5 14 9 10 7 9 8 11 2 3 4 12

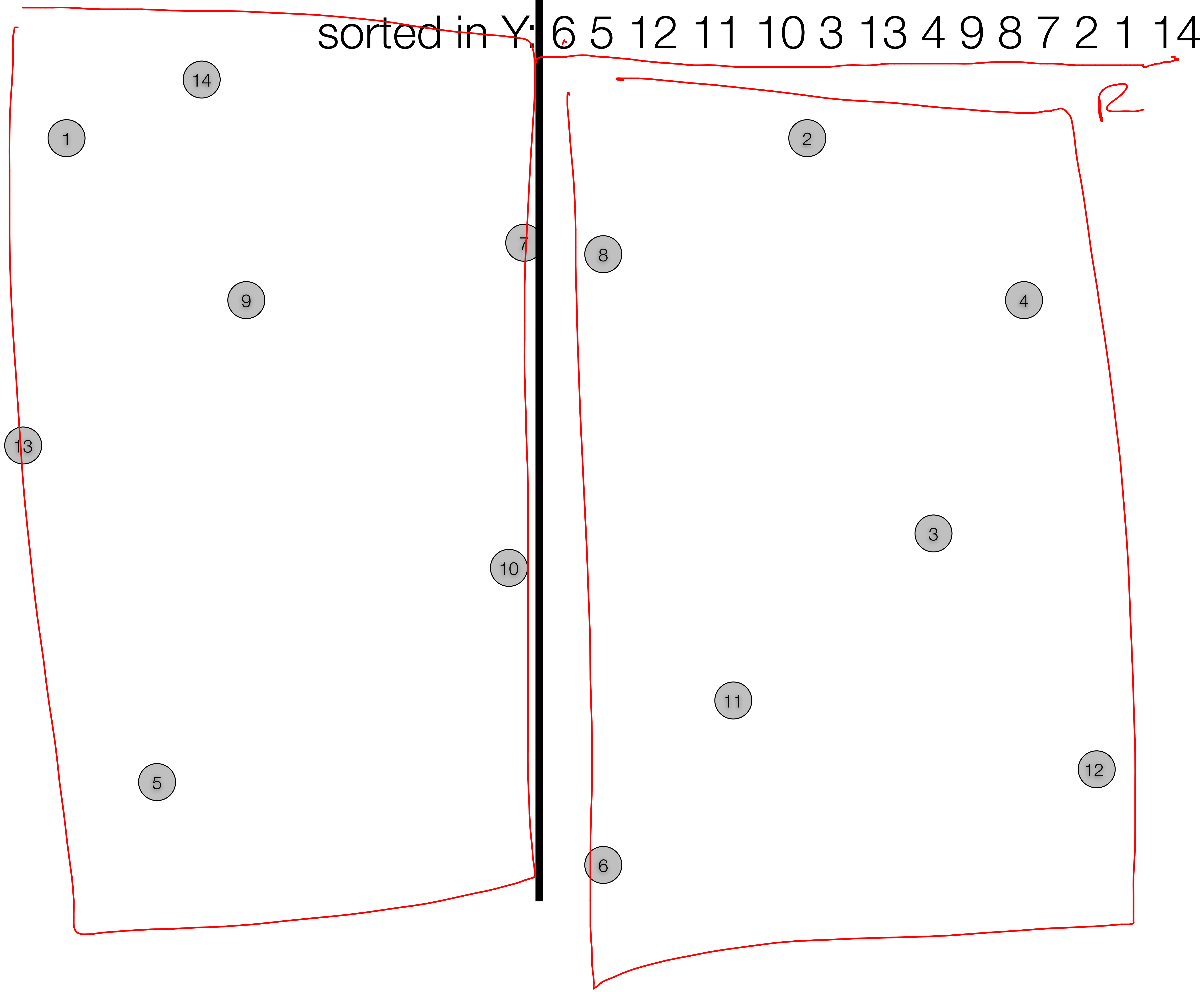
sorted in Y: 6 5 12 11 10 3 13 4 9 8 7 2 1 14



L

sorted in X: 13 1 5 14 9 10 7 9 8 11 2 3 4 12

sorted in Y: 6 5 12 11 10 3 13 4 9 8 7 2 1 14



given S_x, S_y compute

L_x L_y

R_x, R_y

$\Theta(n)$ to produce all 4 of these sets

ClosestPair(P) → array of n points

Compute Sorted-in-X list SX $\Theta(n\log n)$

Compute Sorted-in-Y list SY $\Theta(n\log n)$

Closest(P,SX,SY) → $\Theta(n\log n)$

Total $\Theta(n\log n)$

Closest(P,SX,SY) → BASE CASE when $|P| < 4$, brute force.

Let q be the middle entry of SX

Compute Left, Right, LX, LY, RX, RY based on q

δ, r, j = MIN(closest(L, LX, LY) closest(R, RX, RY))

Mohawk = {SCAN thru SY and choose all points that are within δ of q.x}

For each $x \in$ Mohawk (in order that it was added)

INSpect the next 15 cubbies above or equal in y

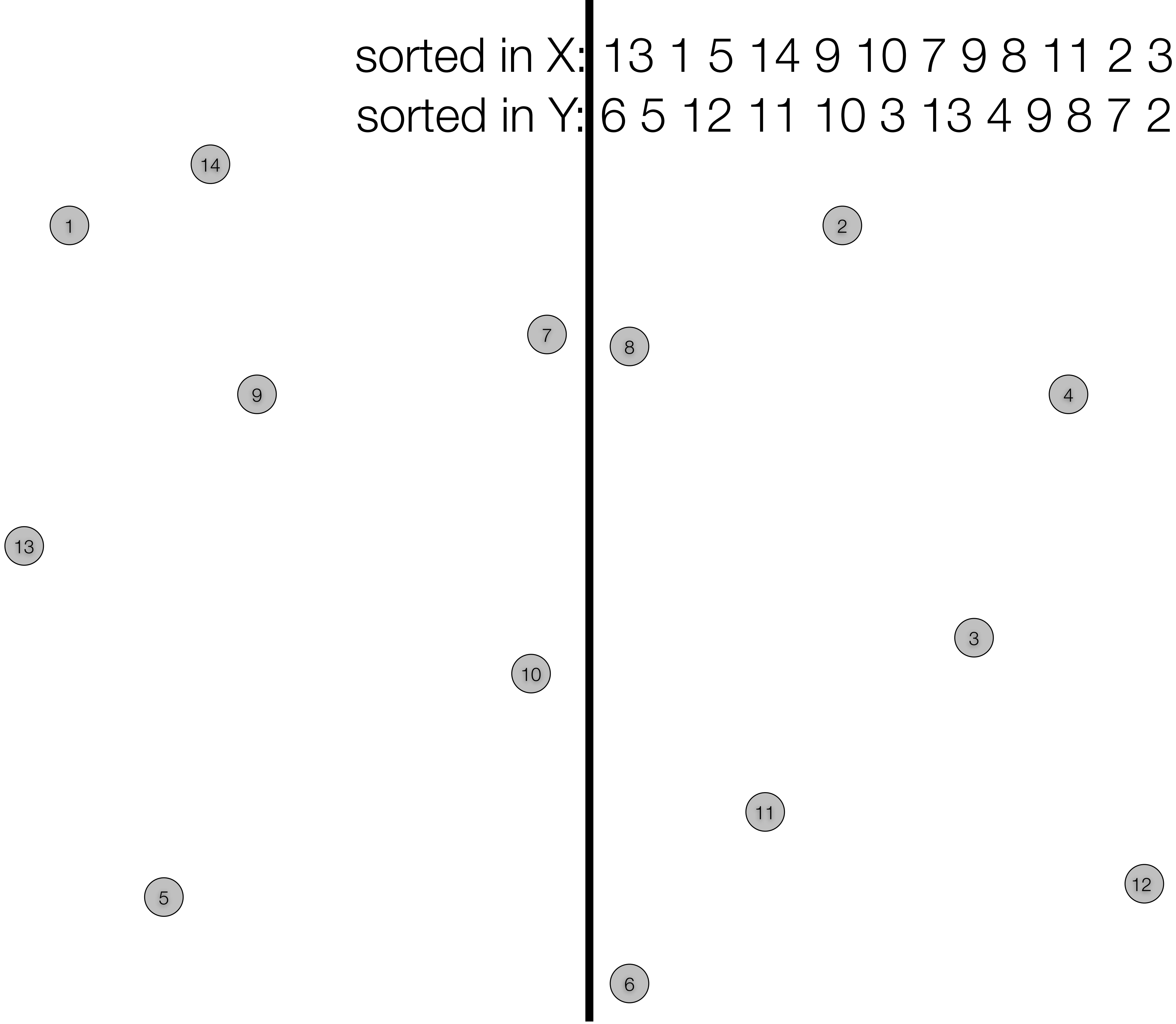
if any cubby contains a point that is closer than δ,

then revise (δ, r, j)

→ Return (δ, r, j)

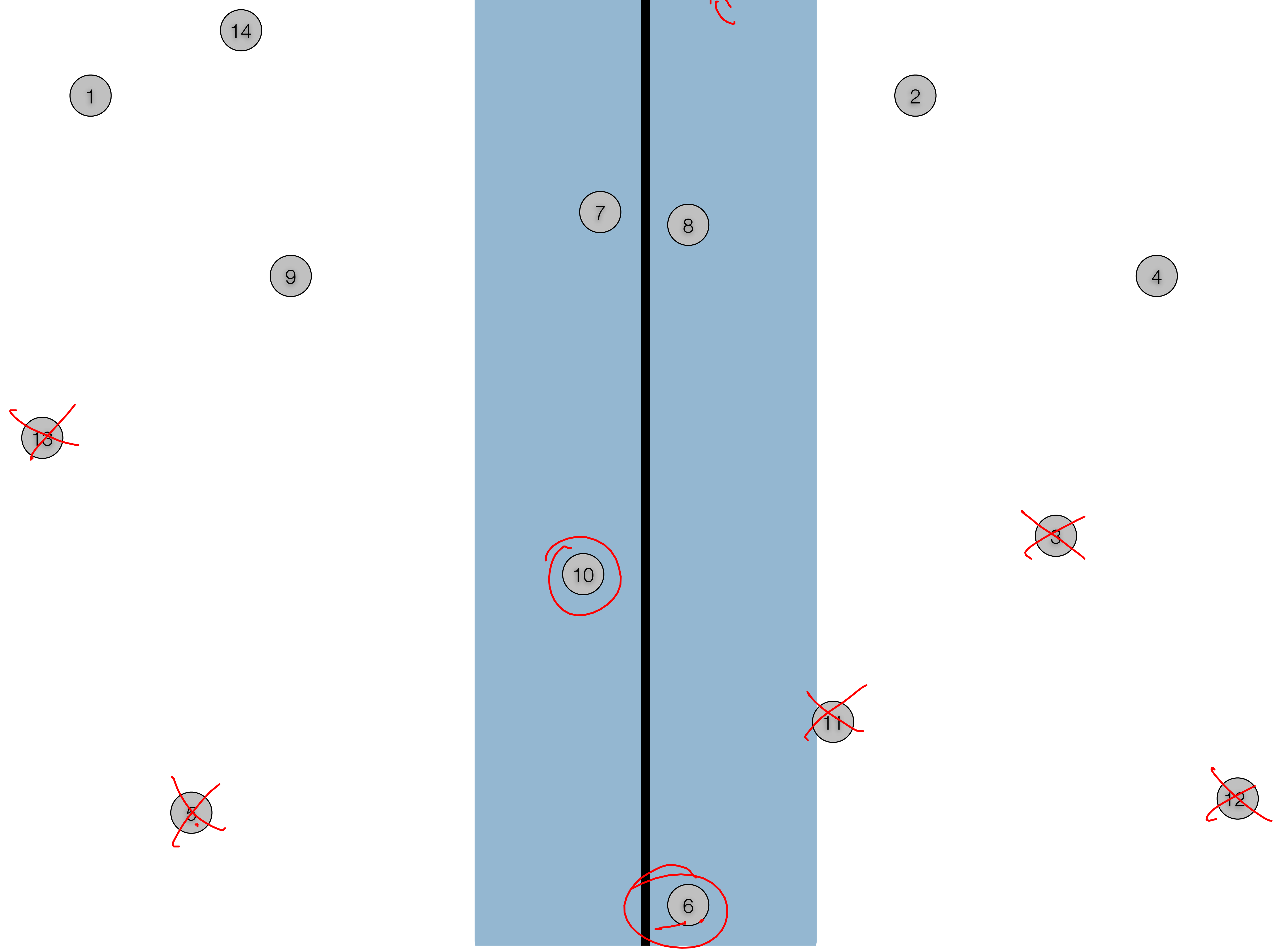
sorted in X: 13 1 5 14 9 10 7 9 8 11 2 3 4 12

sorted in Y: 6 5 12 11 10 3 13 4 9 8 7 2 1 14



sorted in X: 13 1 5 14 9 10 7 9 8 11 2 3 4 12

sorted in Y: 6 5 12 11 10 3 13 4 9 8 7 2 1 14



Closest(P,SX,SY) → BASE CASE

Let q be the middle-element of SX → $\Theta(1)$

Divide P into Left, Right according to q → $\Theta(n)$

delta,r,j = MIN(Closest(Left, LX, LY) Closest(Right, RX, RY)) → $2T(\frac{n}{2}) + \Theta(1)$

$\Theta(1)$

↑ the pair of points that are δ-apart.

Mohawk = { Scan SY, add pts that are delta from q.x } → $\Theta(n)$

For each point x in Mohawk (in order): → $\Theta(n)$ iteration

Compute distance to its next 15 neighbors

Update delta,r,j if any pair (x,y) is < delta

→ each iteration is $\Theta(1)$

Return (delta,r,j)

Total : $T(n) = 2T(\frac{n}{2}) + \Theta(n)$

Closest(P,SX,SY)

Let q be the middle-element of SX

Divide P into Left, Right according to q

delta,r,j = MIN(Closest(Left, LX, LY) Closest(Right, RX, RY))

Mohawk = { Scan SY, add pts that are delta from q.x }

For each point x in Mohawk (in order):

Compute distance to its next 15 neighbors

Update delta,r,j if any pair (x,y) is < delta

Return (delta,r,j)

Can be reduced to 7!

Running time for Closest pair algorithm

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$$

$$\Theta(n \log n)$$

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

Matrix
multiplication

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \bigstar \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 1\cdot 5 + 2\cdot 7 & 1\cdot 6 + 2\cdot 8 \\ 3\cdot 5 + 4\cdot 7 & 3\cdot 6 + 4\cdot 8 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \star \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 5+14 & 6+16 \\ 15+28 & 18+32 \end{bmatrix}$$

$$= \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

$$
\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix}
\begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,n} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,n} \\ \vdots \\ b_{n,1} & b_{n,2} & \cdots & b_{n,n} \end{bmatrix}
=
\begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,n} \end{bmatrix}
$$

$$C_{ij} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj}$$

- each C_{ij} requires $\Theta(n)$

- there are n^2 such elements, so naive alg is $\Theta(n^3)$

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & & & \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix} \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,n} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,n} \\ \vdots & & & \\ b_{n,1} & b_{n,2} & \cdots & b_{n,n} \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & & & \\ c_{n,1} & c_{n,2} & \cdots & c_{n,n} \end{bmatrix}$$

$$c_{i,j} = \sum_{k=1}^{n} a_{i,k} \cdot b_{k,j}$$

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & & & \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix} \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,n} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,n} \\ \vdots & & & \\ b_{n,1} & b_{n,2} & \cdots & b_{n,n} \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & & & \\ c_{n,1} & c_{n,2} & \cdots & c_{n,n} \end{bmatrix}$$

4 parts that are

$n/2 \times n/2$

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}\begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix}$$

$\frac{n}{2} \times \frac{n}{2}$

$$\underbrace{T(n) = \overset{a}{8}\,T\!\left(\frac{n}{\overset{b}{2}}\right) + \Theta(n^2)}_{} = \Theta(n^{\log_2 8}) = \Theta(n^3)$$

time for an $n \times n$ matrix multiplication

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix}$$

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix}$$

$$T(n) = 8T(n/2) + \Theta(n^2)$$

$$\Theta(n^3)$$

$$= \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix}$$

$P_1 + P_2$

$(AF - \cancel{AH}) + \cancel{AH} + BH$

$P_3 + P_4 = (CE + \cancel{DE}) + (DG - \cancel{DE}) = CE + DG$

[Strassen]

$$P_1 = A(F - H)$$

$$P_2 = (A + B)H$$

$$P_3 = (C + D)E$$

$$P_4 = D(G - E)$$

$$P_5 = (A + D)(E + H)$$

$$P_6 = (B - D)(G + H)$$

$$P_7 = (A - C)(E + F)$$

each is an $\left(\frac{n}{2}\right)$-matrix mult

$$= \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix}$$

[strassen]

$$P_1 = A(F - H)$$

$$P_2 = (A + B)H$$

$$P_3 = (C + D)E$$

$$P_4 = D(G - E)$$

$$P_5 = (A + D)(E + H)$$

$$P_6 = (B - D)(G + H)$$

$$P_7 = (A - C)(E + F)$$

$$= \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix}$$

[strassen]

$$P_1 = A(F - H)$$

$$P_2 = (A + B)H$$

$$P_3 = (C + D)E$$

$$P_4 = D(G - E)$$

$$P_5 = (A + D)(E + H)$$

$$P_6 = (B - D)(G + H)$$

$$P_7 = (A - C)(E + F)$$

$$= \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix}$$

$R = P_5 + P_4 - P_2 + P_6$

$S = P_1 + P_2$

$T = P_3 + P_4$

$U = P_5 + P_1 - P_3 - P_7$

[strassen]

$P_1 = A(F - H)$

$P_2 = (A + B)H$

$P_3 = (C + D)E$

$P_4 = D(G - E)$

$P_5 = (A + D)(E + H)$

$P_6 = (B - D)(G + H)$

$P_7 = (A - C)(E + F)$

$$\overbrace{(AE + \cancel{AH} + \cancel{DE} + \cancel{DH})}^{P5} + \overbrace{(\cancel{DG} - \cancel{DE})}^{P4} - \overbrace{(\cancel{AH} + \cancel{BH})}^{P2} + \overbrace{(BG + \cancel{BH} - \cancel{DG} - \cancel{DH})}^{P6}$$

$$\rightarrow T(n) = 7T\left(\frac{n}{2}\right) + 18\,\Theta(n^2)$$

$$= \Theta\left(n^{\log_2 7}\right) \sim n^{2.81}$$

$$= \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix}$$

$$R = P_5 + P_4 - P_2 + P_6 \qquad S = P_1 + P_2$$

$$T = P_3 + P_4 \qquad U = P_5 + P_1 - P_3 - P_7$$

[strassen]

$$P_1 = A(F - H)$$

$$P_2 = (A + B)H$$

$$P_3 = (C + D)E$$

$$P_4 = D(G - E)$$

$$P_5 = (A + D)(E + H)$$

$$P_6 = (B - D)(G + H)$$

$$P_7 = (A - C)(E + F)$$

$$M(n) = 7M(n/2) + 18n^2$$

$$= \Theta(n^{\log_2 7}) \sim n^{2.81}$$

taking this idea further

3x3 matricies

check

$$T(n) = \underline{\underline{21}}\, T(n/3) + \Theta(n^2)$$

$$= \Theta(n^{\log_3 23}) = \sim n^{2.77}$$

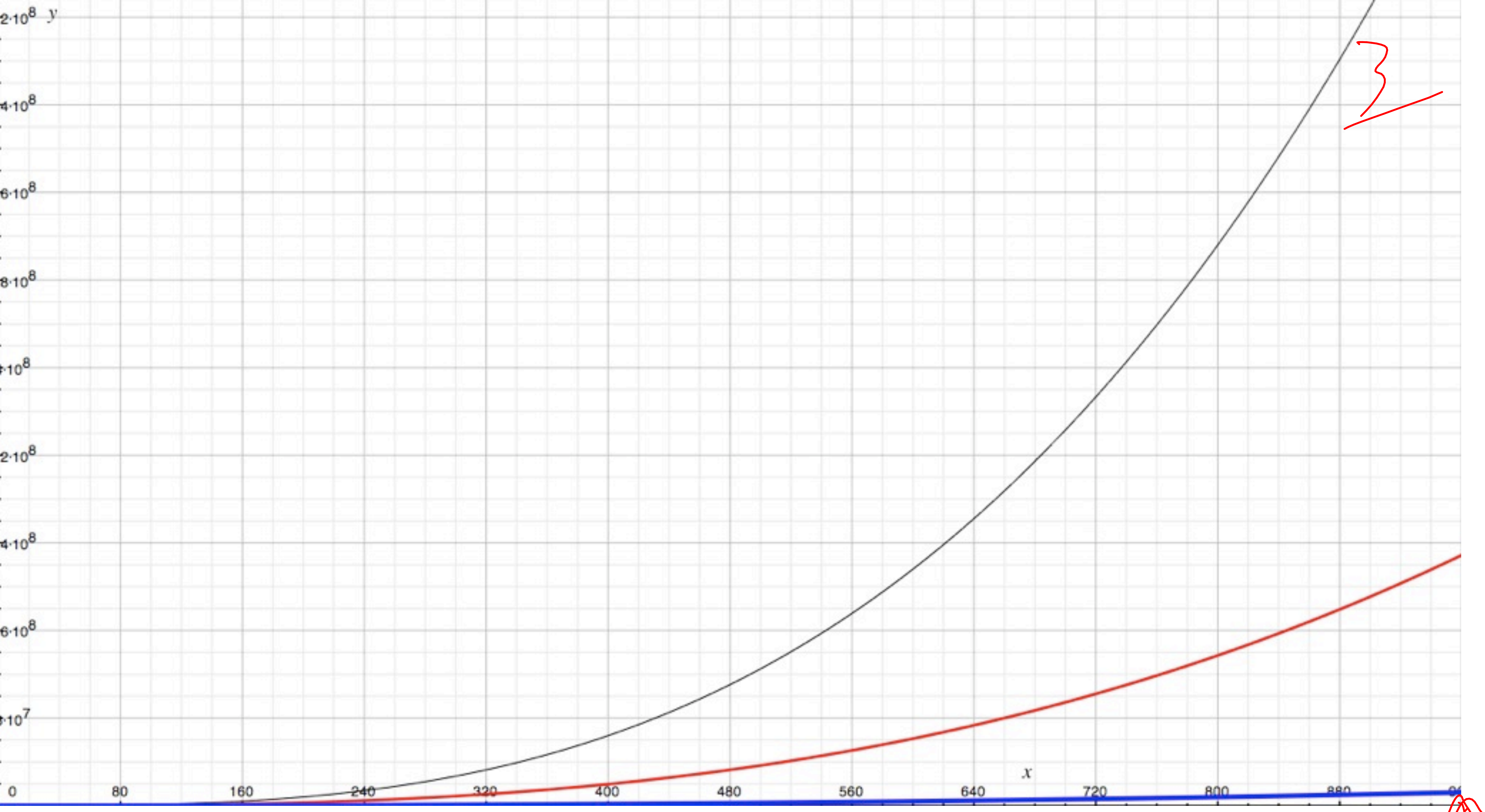
1978 victor pan method

$\underbrace{70\text{x}70}$ matrix using $\underbrace{143640}$ mults

what is the recurrence:

$$T(n) = 143640\, T(n/70) + \Theta(n^2)$$

$$\Theta\left(n^{\log_{70} 143640}\right) = \sim n^{2.795}$$

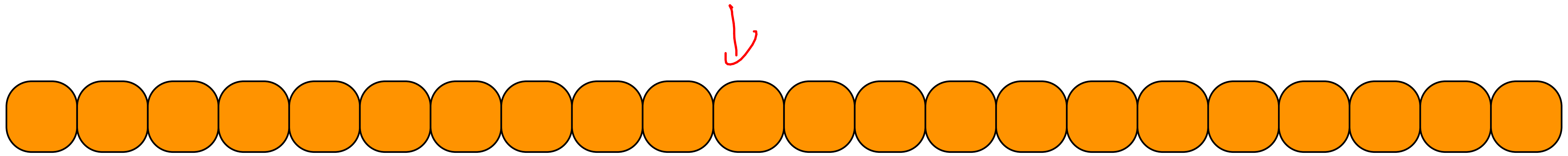


2.81

$O(2.36)$

$\Omega(n^2)$

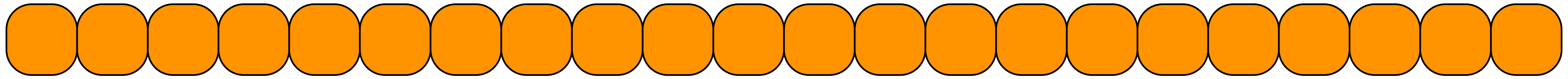
MEDIAN



problem: given a list of n elements, find the element of rank $n/2$. (half are larger, half are smaller)

can sort the list

$\Theta(n \log n)$

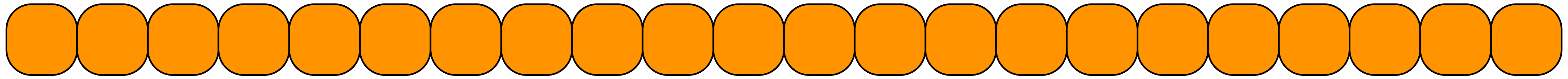


problem: given a list of n elements, find the element of rank $n/2$. (half are larger, half are smaller)

can generalize to i

first solution: sort and pluck.

$$O(n \log n)$$

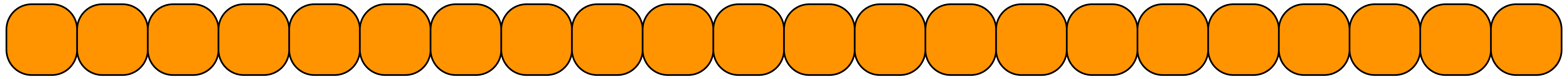


problem: given a list of n elements, find the element of rank i.

$n/2$

ABSTRACTION: find the i^{th} element in RANK

$i = \frac{n}{2}$ corresponds to MEDIAN.

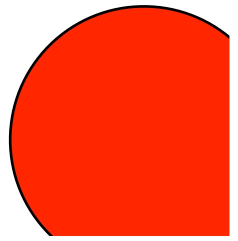


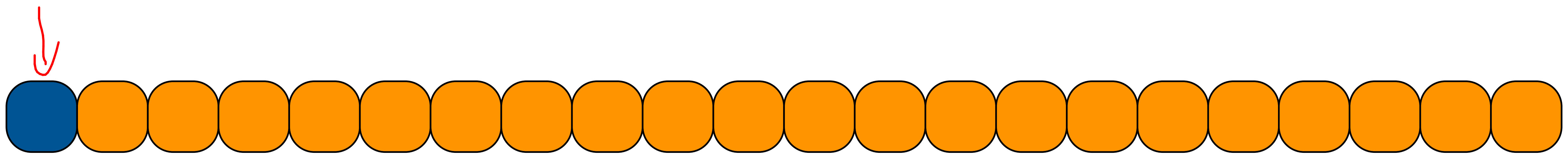
problem: given a list of n elements, find the element of rank i.

key insight:

we do not have to “fully” sort.

semi sort can suffice.

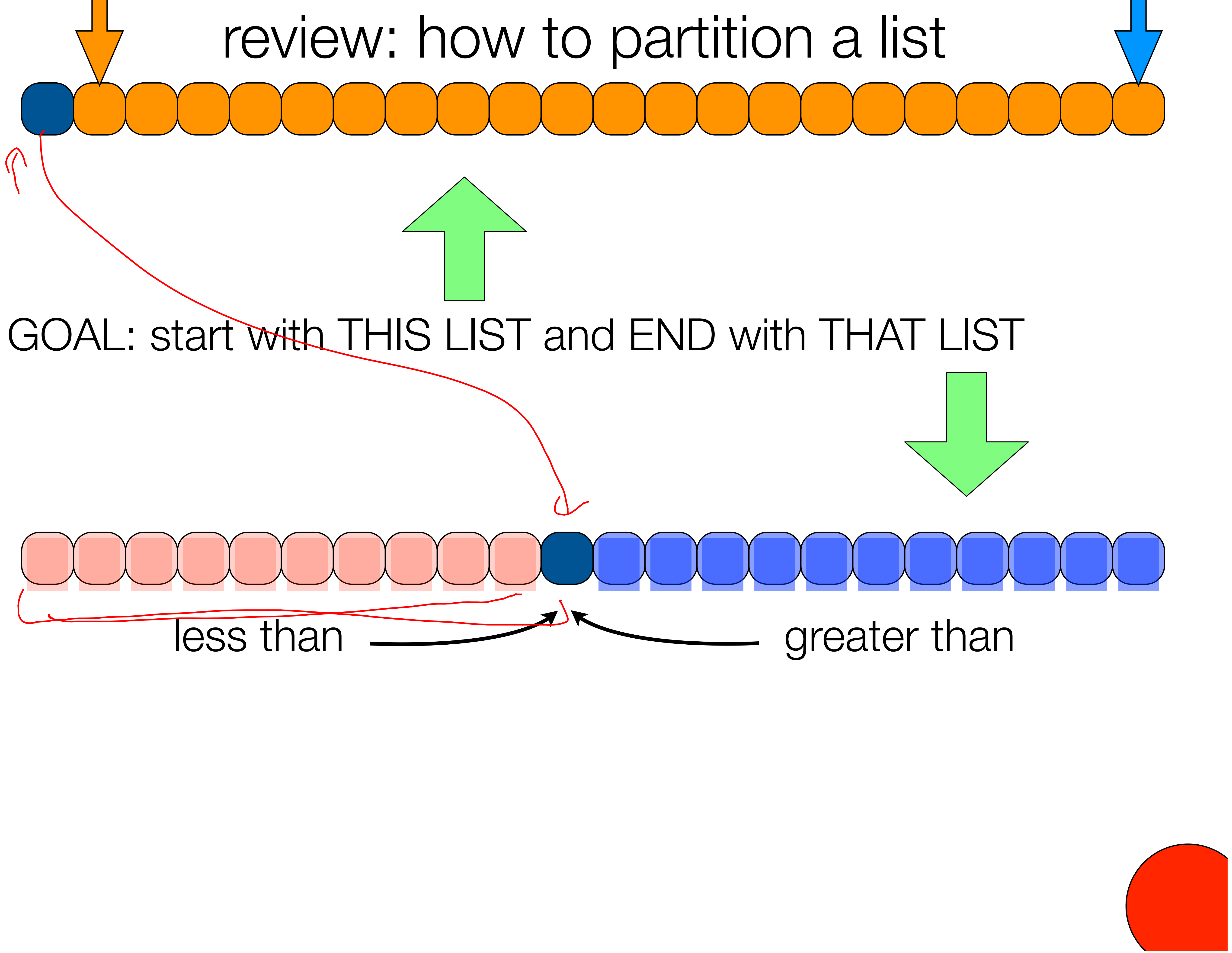




pick first element
partition list about this one
see where we stand

review: how to partition a list

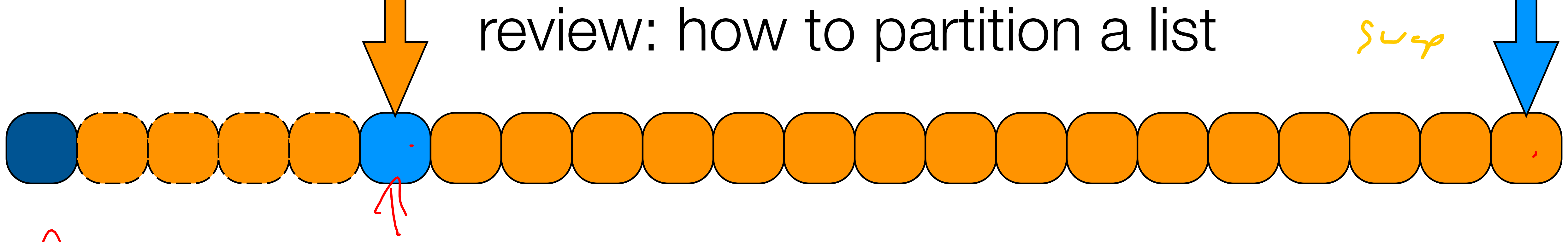
review: how to partition a list



review: how to partition a list

review: how to partition a list

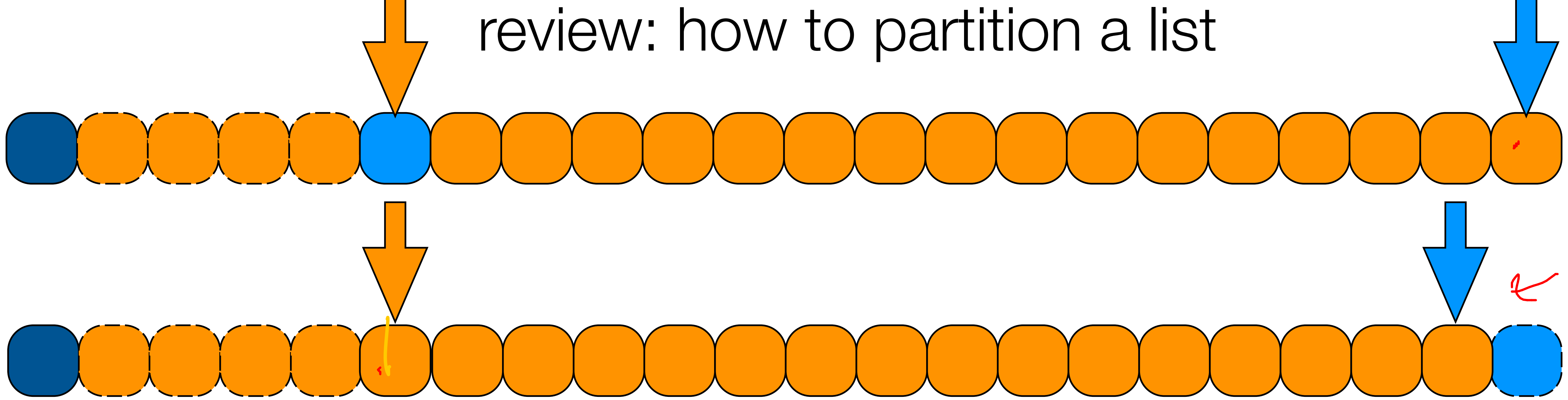
swap



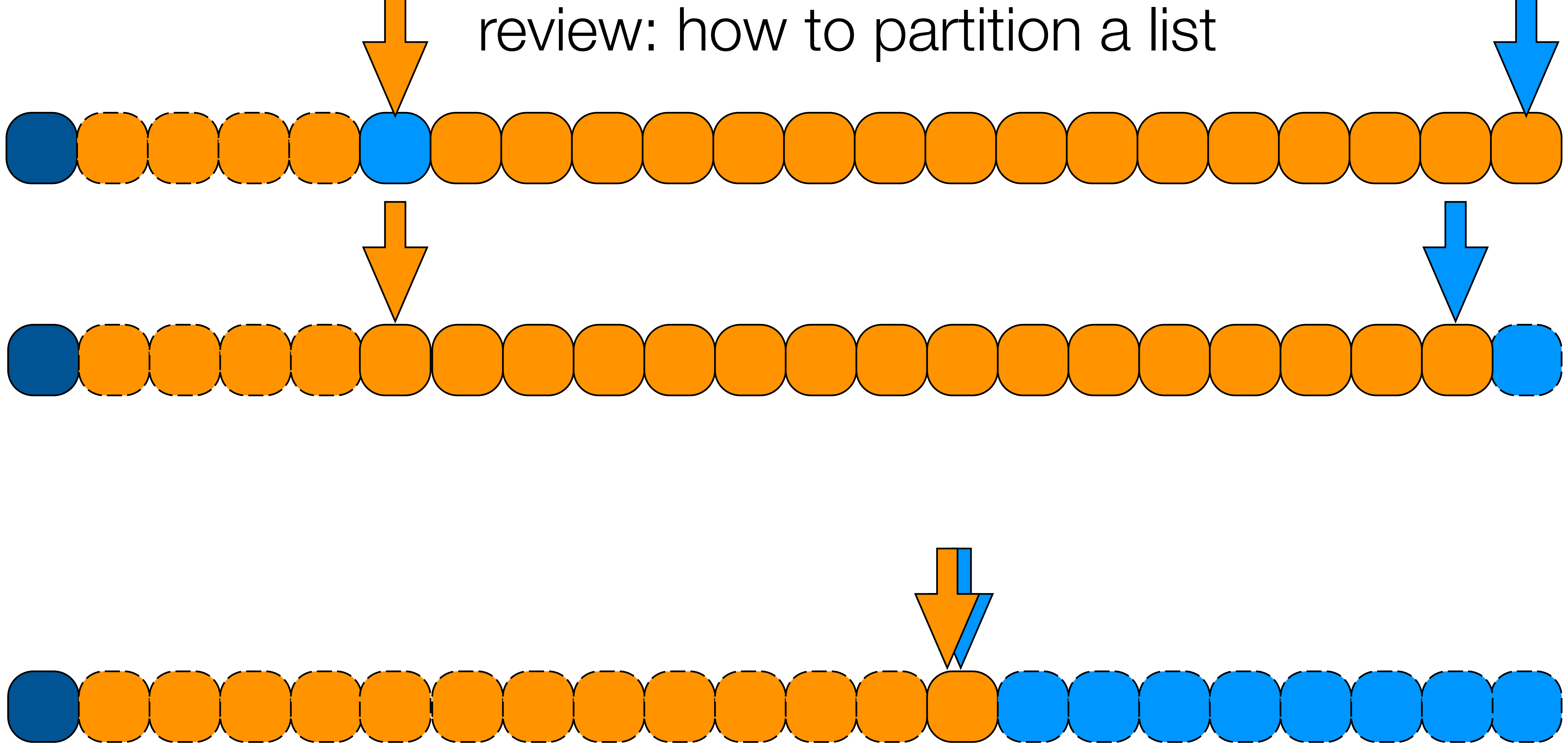
① first element that is larger than

② swap w/ back ptr

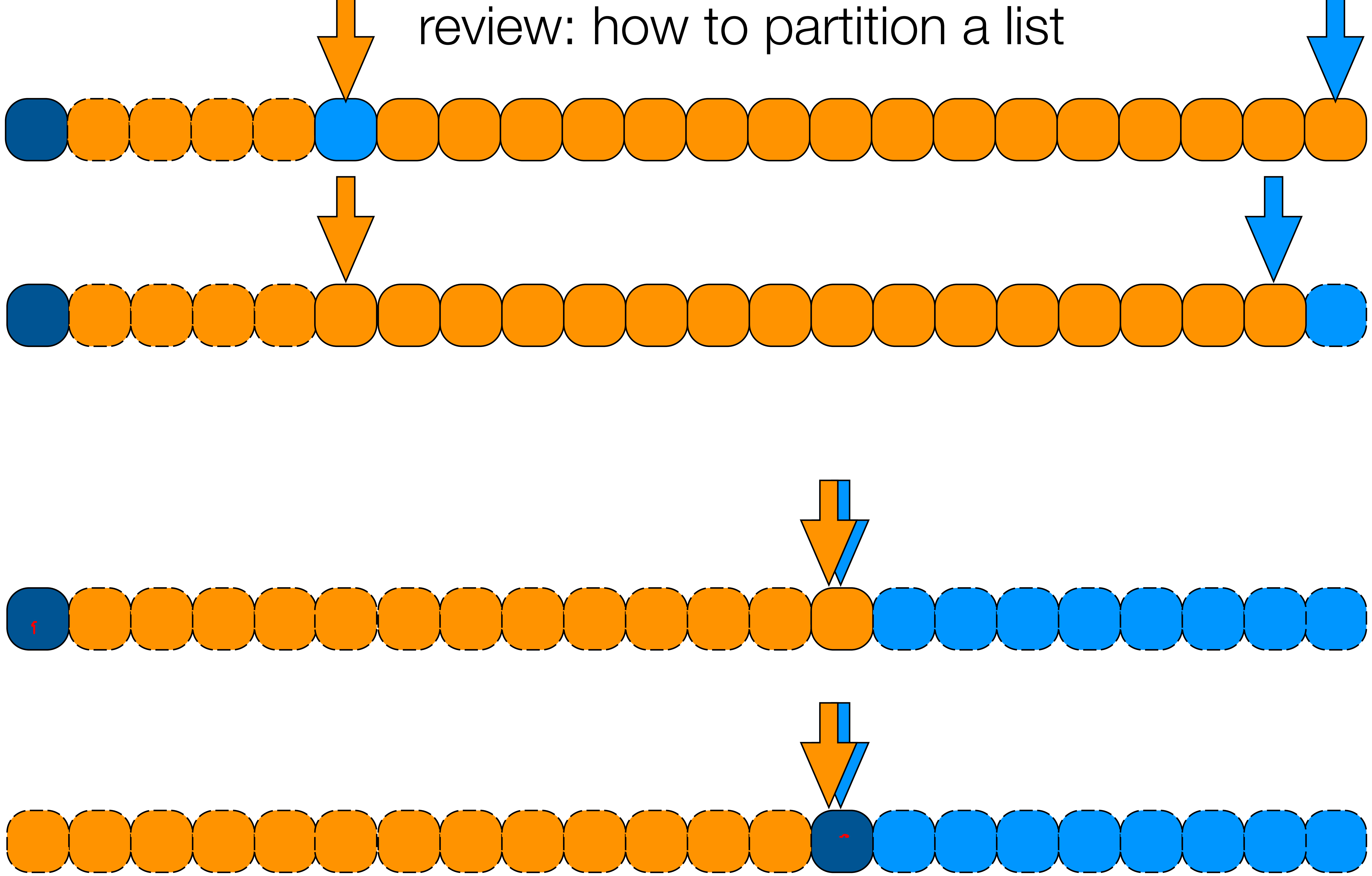
review: how to partition a list



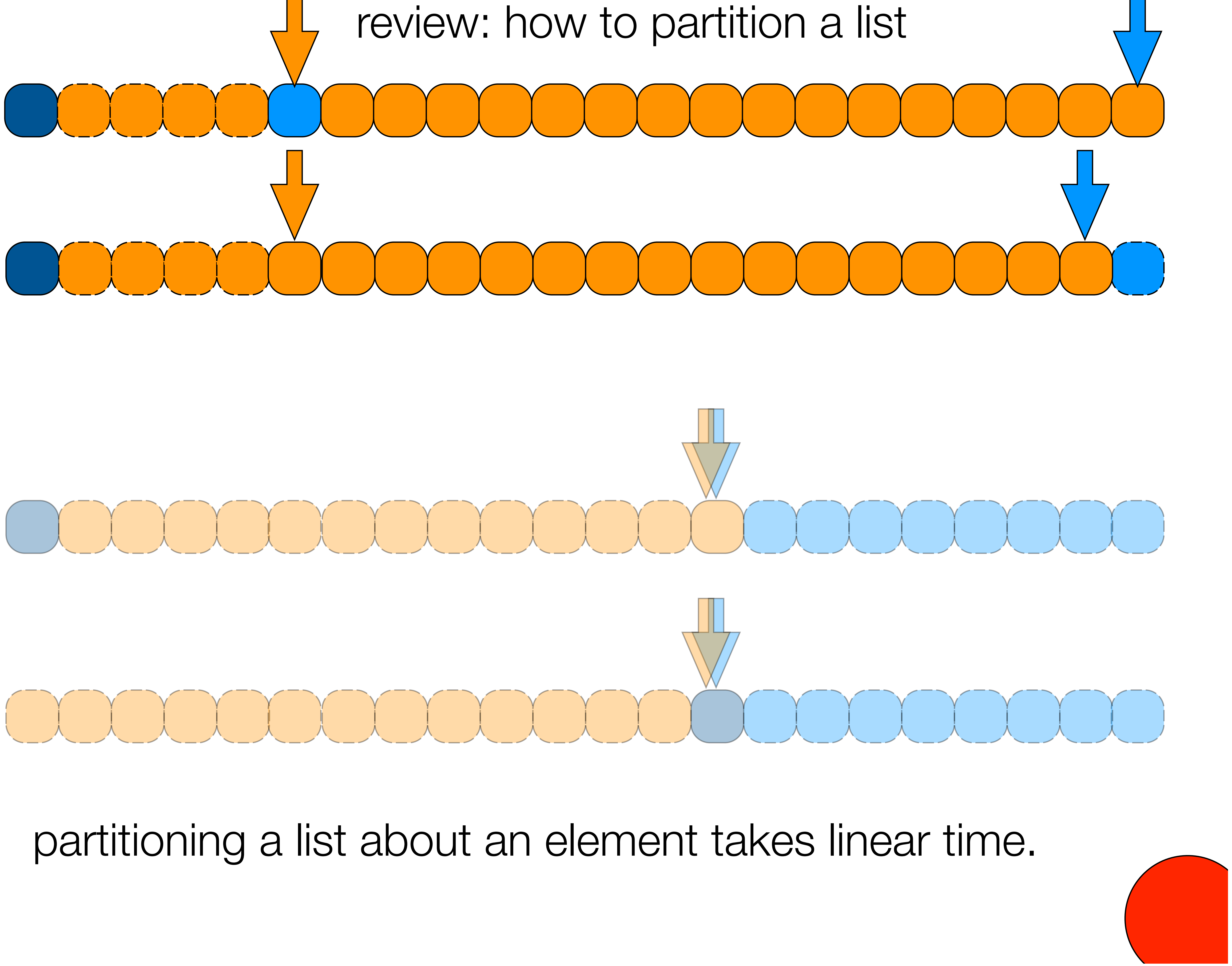
review: how to partition a list



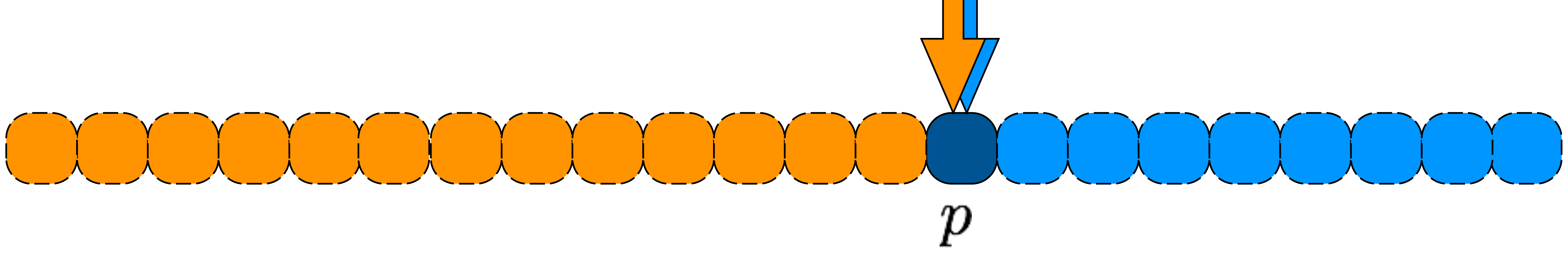
review: how to partition a list



review: how to partition a list



partitioning a list about an element takes linear time.



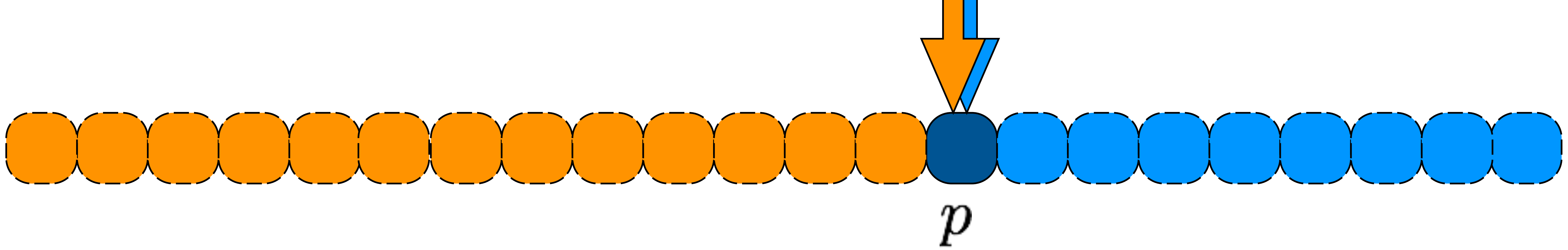
select $(i, A[1, \ldots, n])$

(1) partition using the first element

(2) if rank i < rank p, select$(i, A[1 \ldots p])$

(3) if rank i > rank p, select$(i-p, A[p, \ldots n])$

(4) if rank i = rank p = return p.



select $(i, A[1, \ldots, n])$

handle base case.

partition list about first element

if pivot is position i, return pivot

$\Theta(n)$

else if pivot is in position $> i$ select $(i, A[1, \ldots, p-1])$

else select $((i-p-1), A[p+1, \ldots, n])$

Lets imagine that each partition splits the array into 2 parts of size $n/2$

$$T(n) = T\left(\frac{n}{2}\right) + \Theta(n) = \Theta(n)$$

select $(i, A[1, \ldots, n])$

Assume our partition always splits list into two eql parts

handle base case.

partition list about first element

if pivot is position i, return pivot

else if pivot is in position $> i$ select $(i, A[1, \ldots, p-1])$

else select $((i-p-1), A[p+1, \ldots, n])$

select $(i, A[1, \ldots, n])$

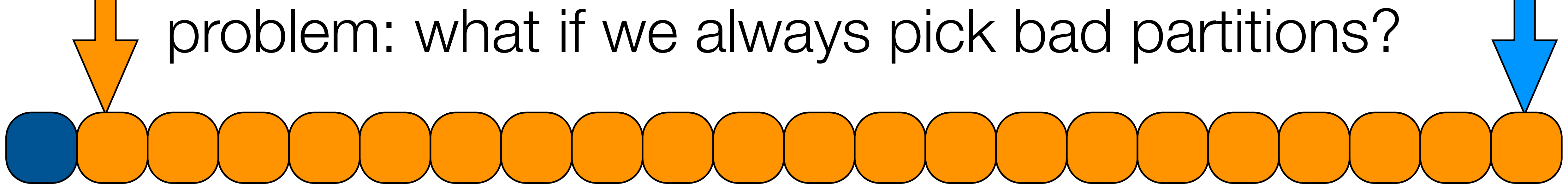
Assume our partition always splits list into two eql parts

handle base case.
partition list about first element
if pivot is position i, return pivot
else if pivot is in position $> i$ select $(i, A[1, \ldots, p-1])$
else select $((i-p-1), A[p+1, \ldots, n])$

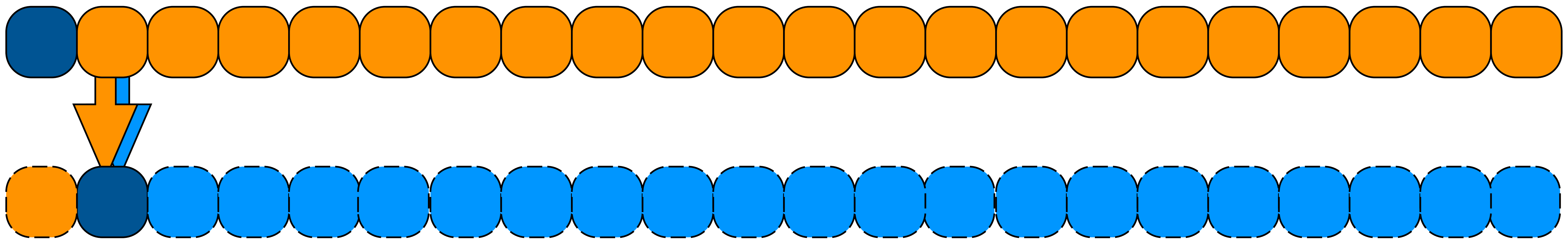
$$T(n) = T(n/2) + O(n)$$

$$\Theta(n)$$

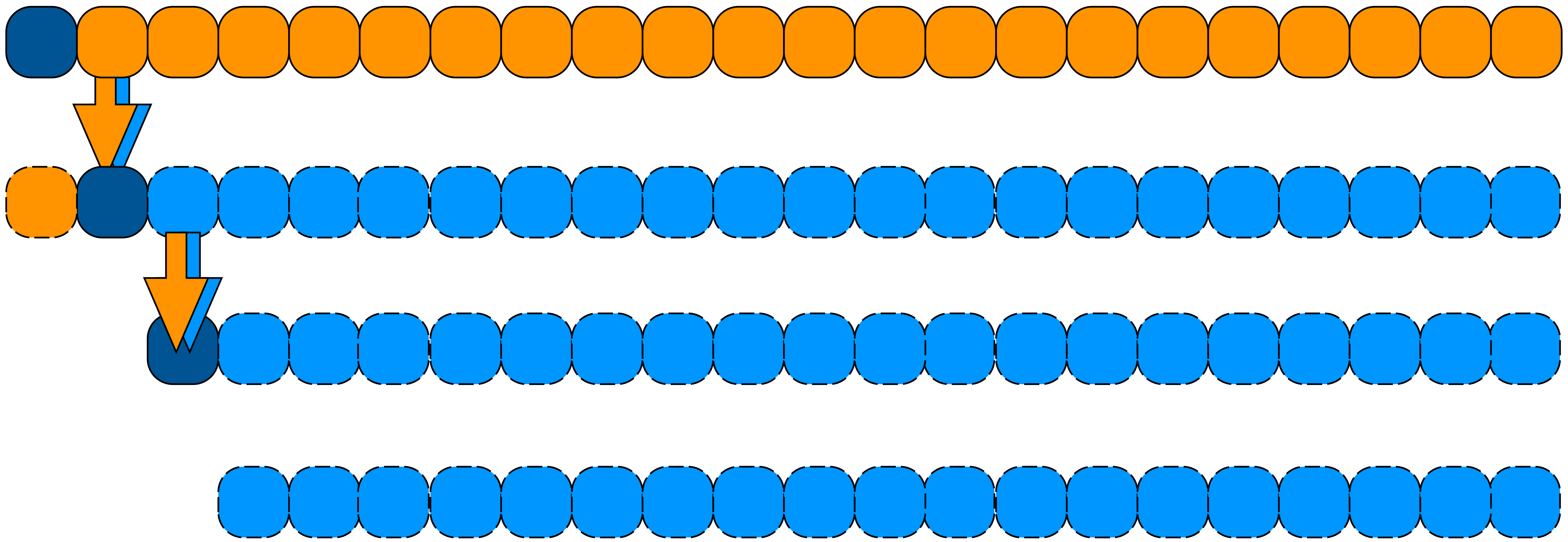
problem: what if we always pick bad partitions?

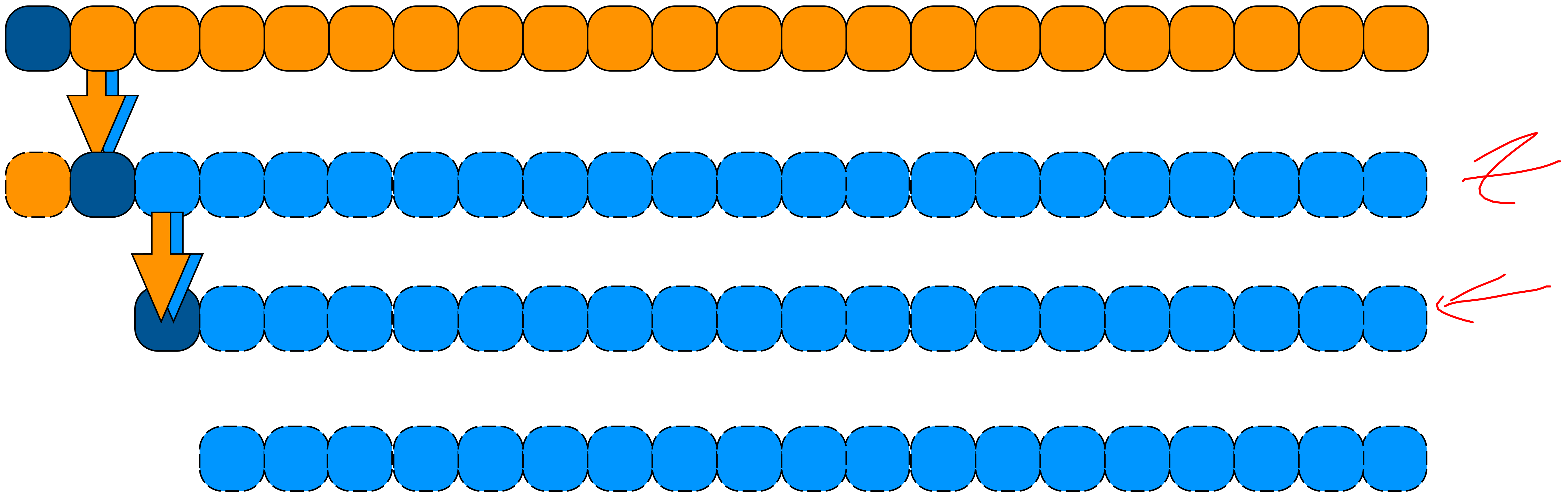


problem: what if we always pick bad partitions?



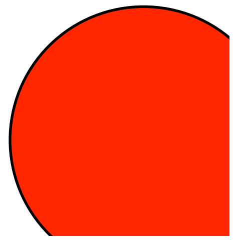
problem: what if we always pick bad partitions?





problem: what if we always pick bad partitions?

$\Rightarrow O(n^2)$



select $(i, A[1, \ldots, n])$

handle base case.

partition list about first element

if pivot is position i, return pivot

else if pivot is in position $> i$ select $(i, A[1, \ldots, p-1])$

else select $((i-p-1), A[p+1, \ldots, n])$

select $(i, A[1, \ldots, n])$

handle base case.

partition list about first element

if pivot is position i, return pivot

else if pivot is in position $> i$ select $(i, A[1, \ldots, p-1])$

else select $((i-p-1), A[p+1, \ldots, n])$

$$T(n) = T(n-1) + O(n)$$

$$\Theta(n^2)$$

Needed:

a good partition element

partition $(A[1, \ldots, n])$

Needed:

a good partition element

partition $(A[1,\ldots,n])$

produce an element where
30% smaller, 30% larger

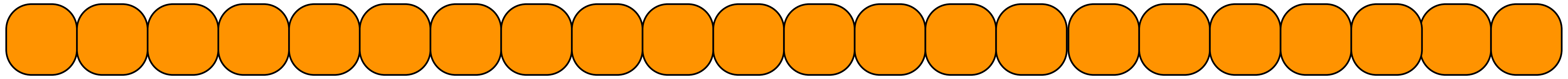
solution:
bootstrap

image: gucci

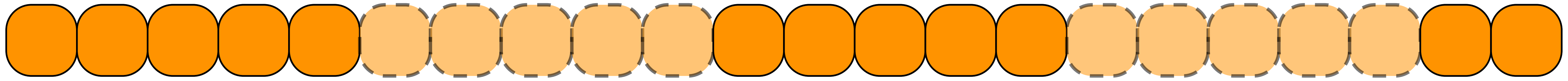
image: mark nason

image: d&g

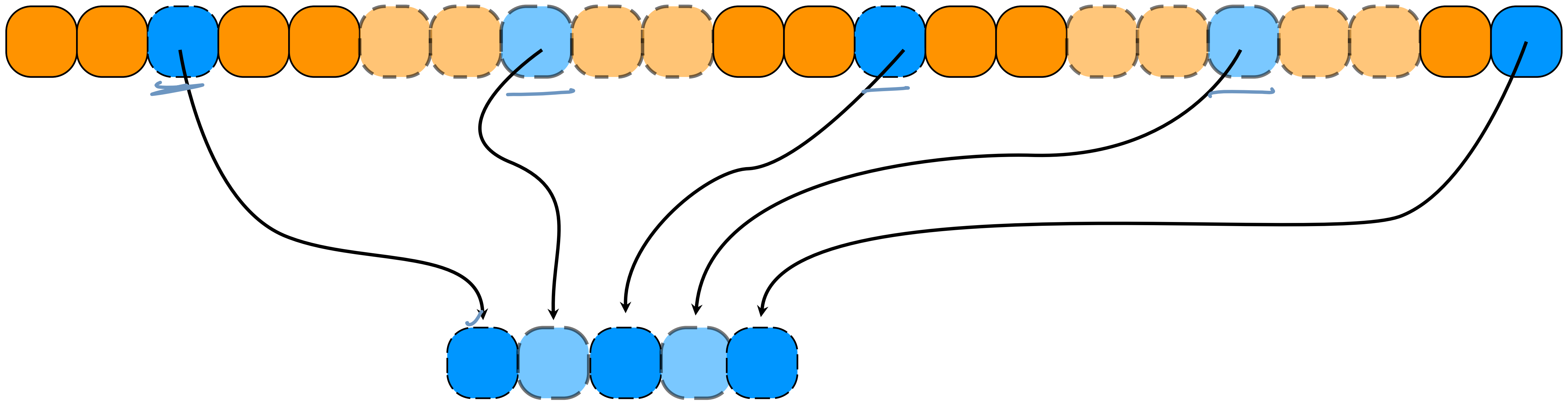
partition $(A[1, \ldots, n])$

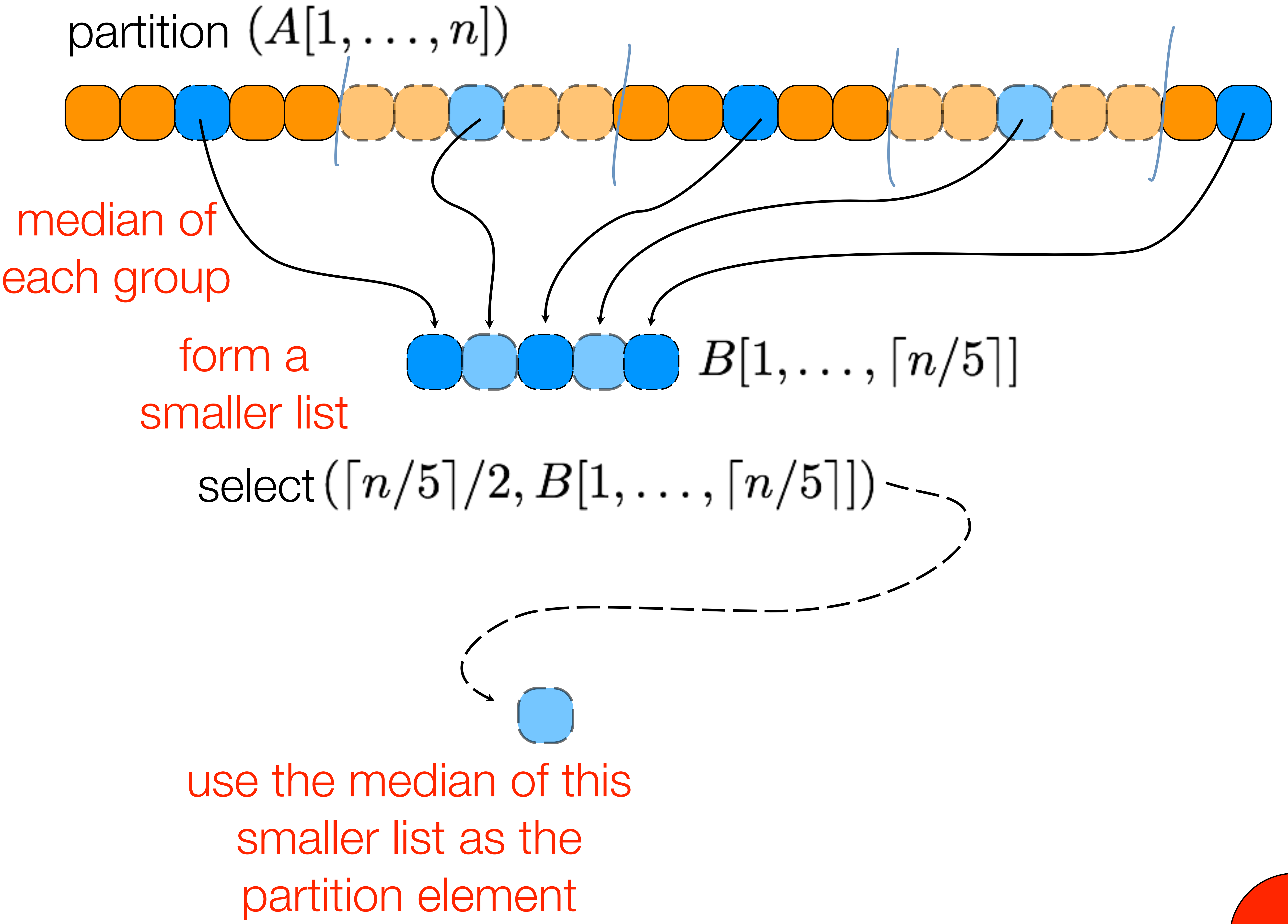


partition $(A[1, \ldots, n])$

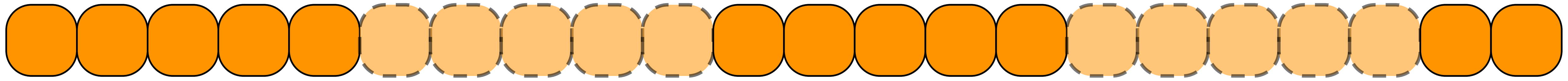


partition $(A[1, \ldots, n])$



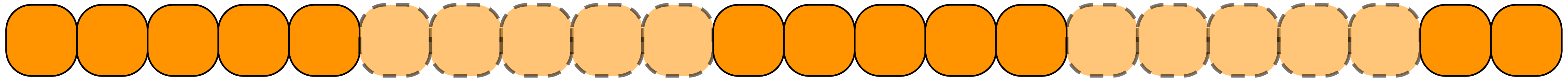

partition $(A[1,\ldots,n])$
median of each group
form a smaller list
$B[1,\ldots,\lceil n/5\rceil]$
select $(\lceil n/5\rceil/2, B[1,\ldots,\lceil n/5\rceil])$
use the median of this smaller list as the partition element

partition $(A[1, \ldots, n])$



1.
2.
3.
4.
5.

partition $(A[1, \ldots, n])$



divide list into groups of 5 elements

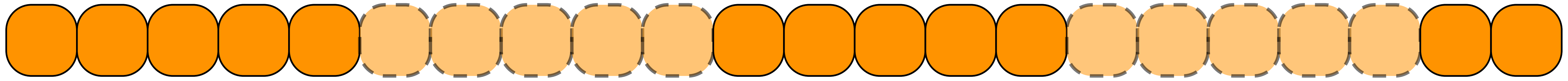
find median of each small list

gather all medians

call select(...) on this sublist to find median

return the result

partition $(A[1,\ldots,n])$



divide list into groups of 5 elements

find median of each small list

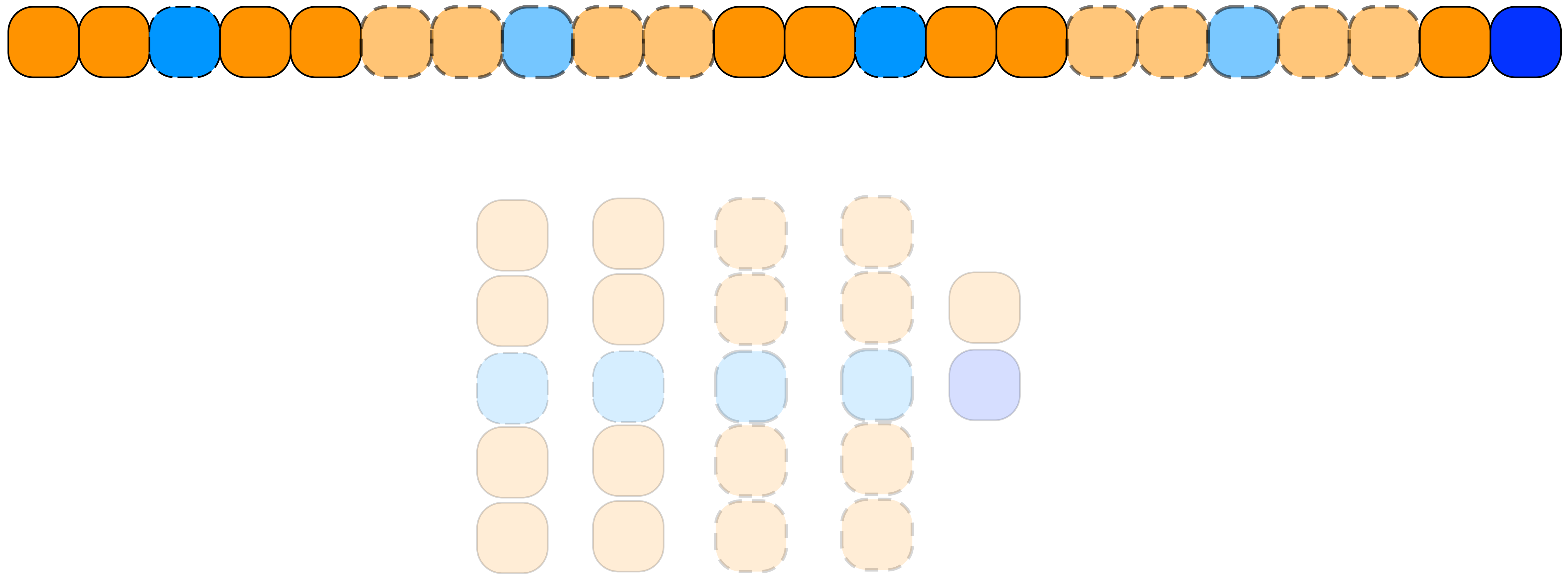
gather all medians

call select(...) on this sublist to find median

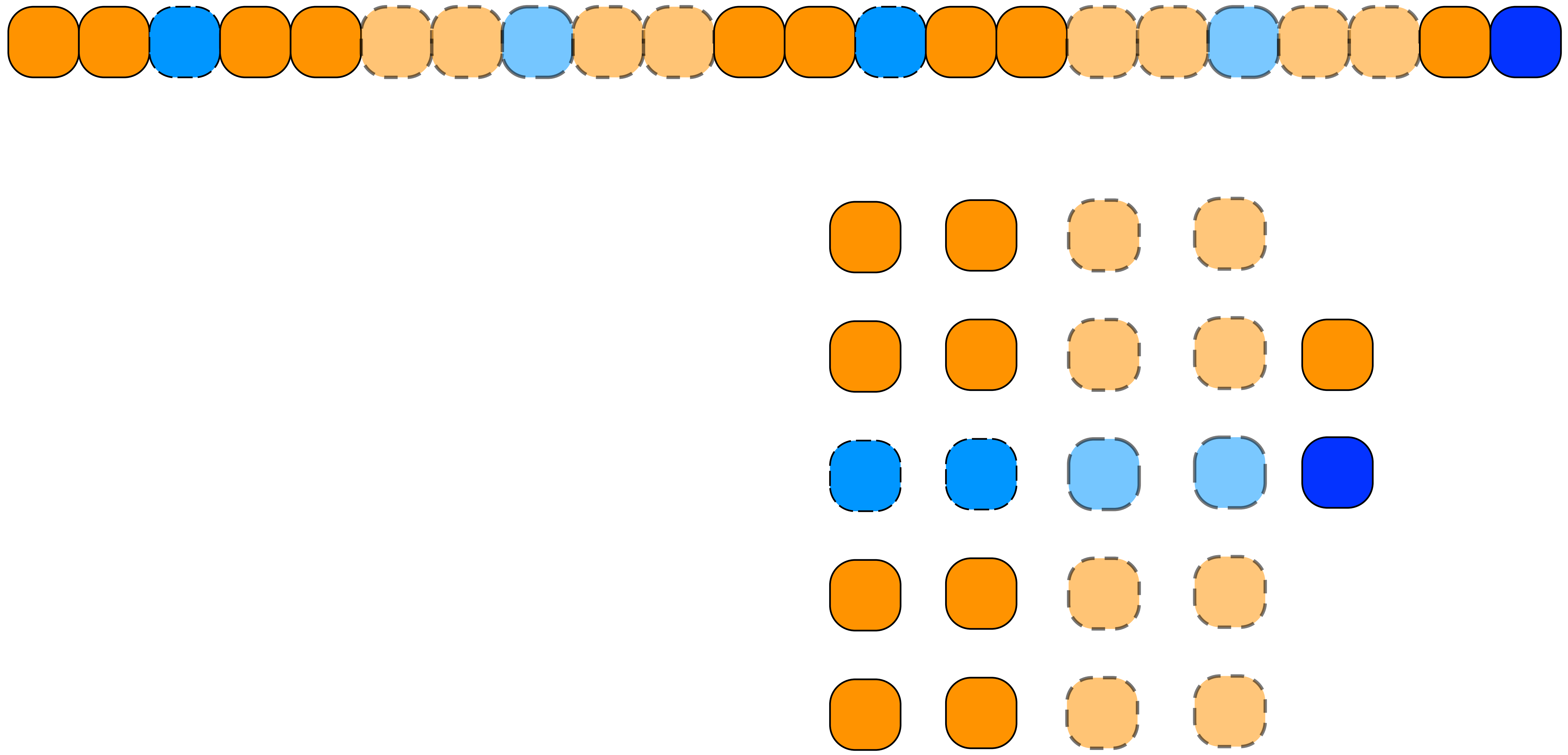
return the result

$$P(n) = S(\lceil n/5 \rceil) + O(n)$$

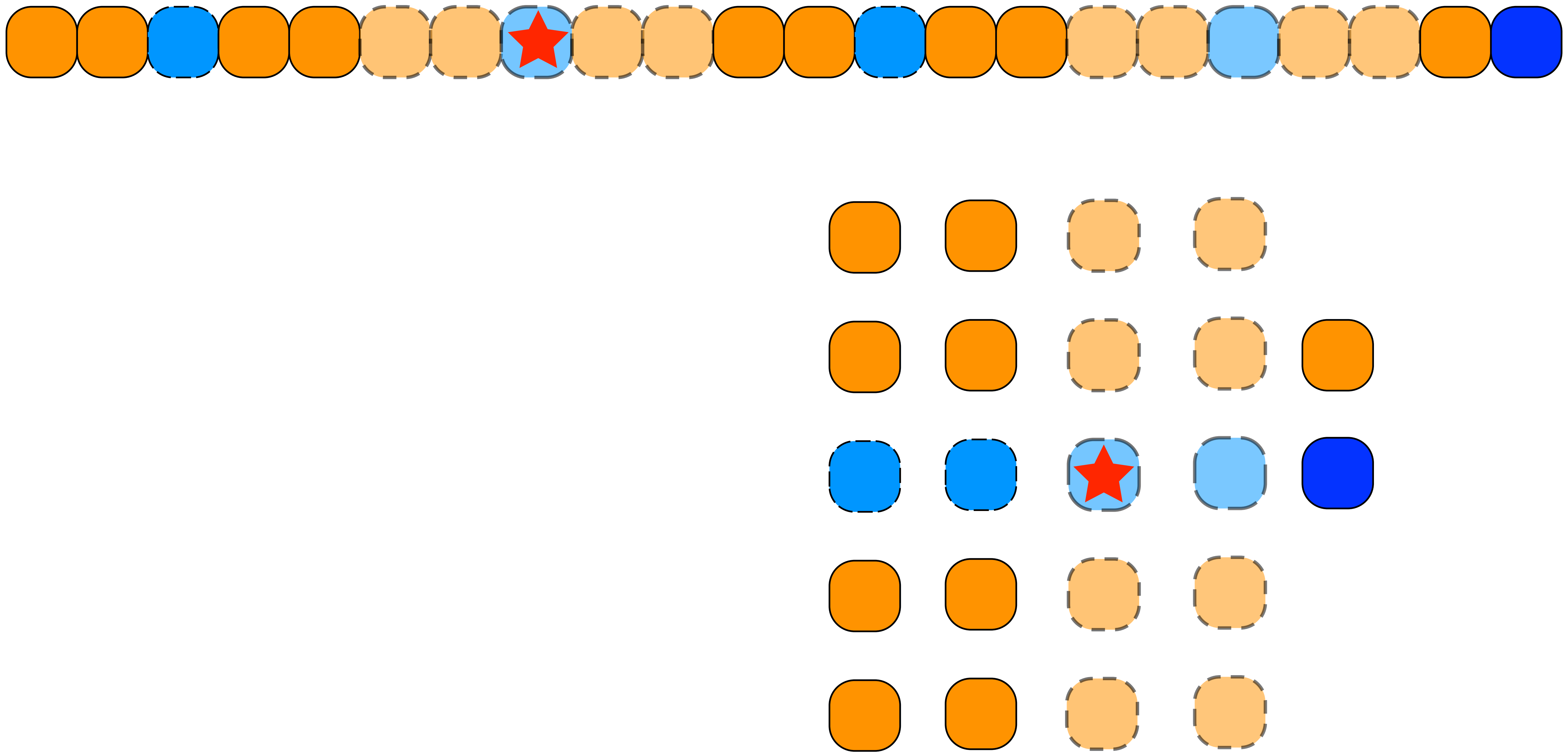
a nice property of our partition



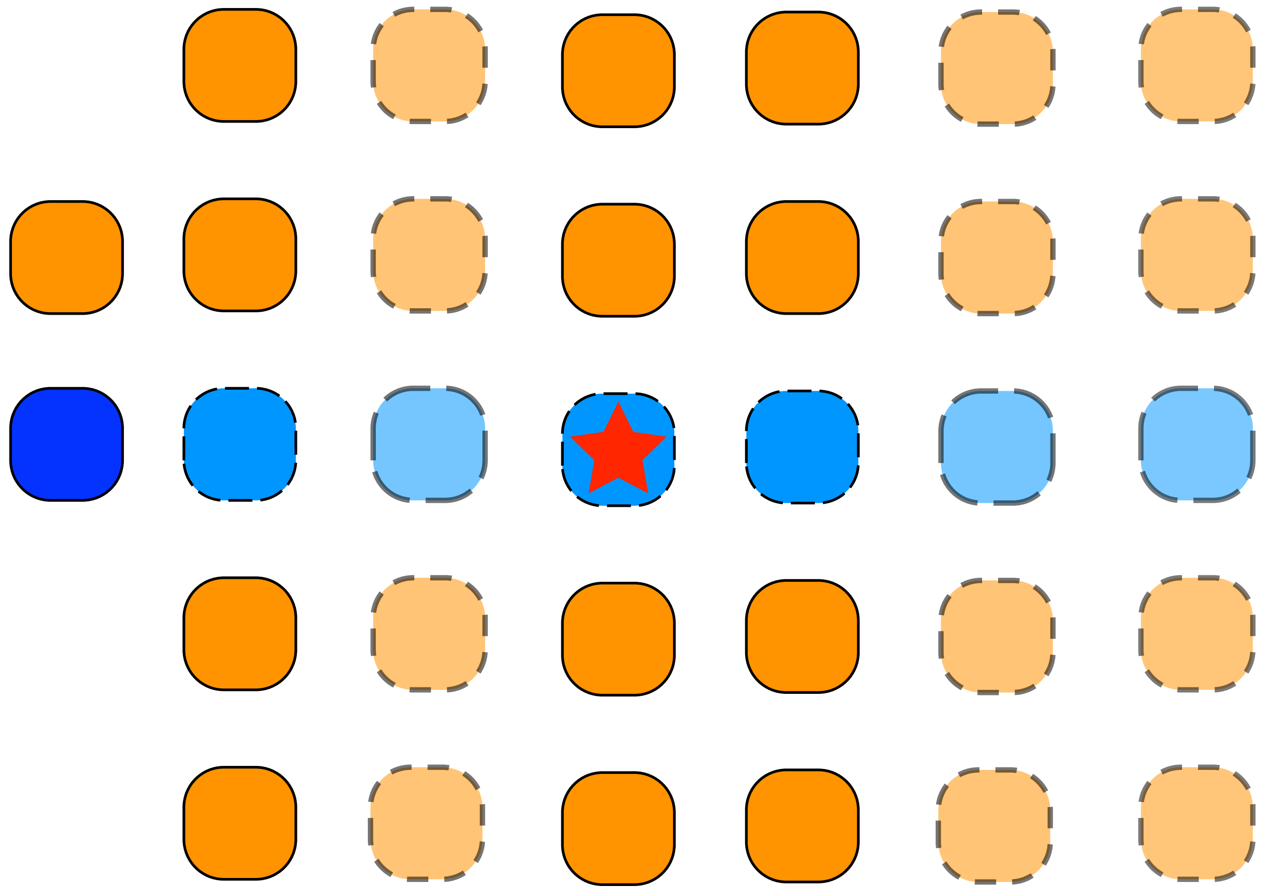
a nice property of our partition



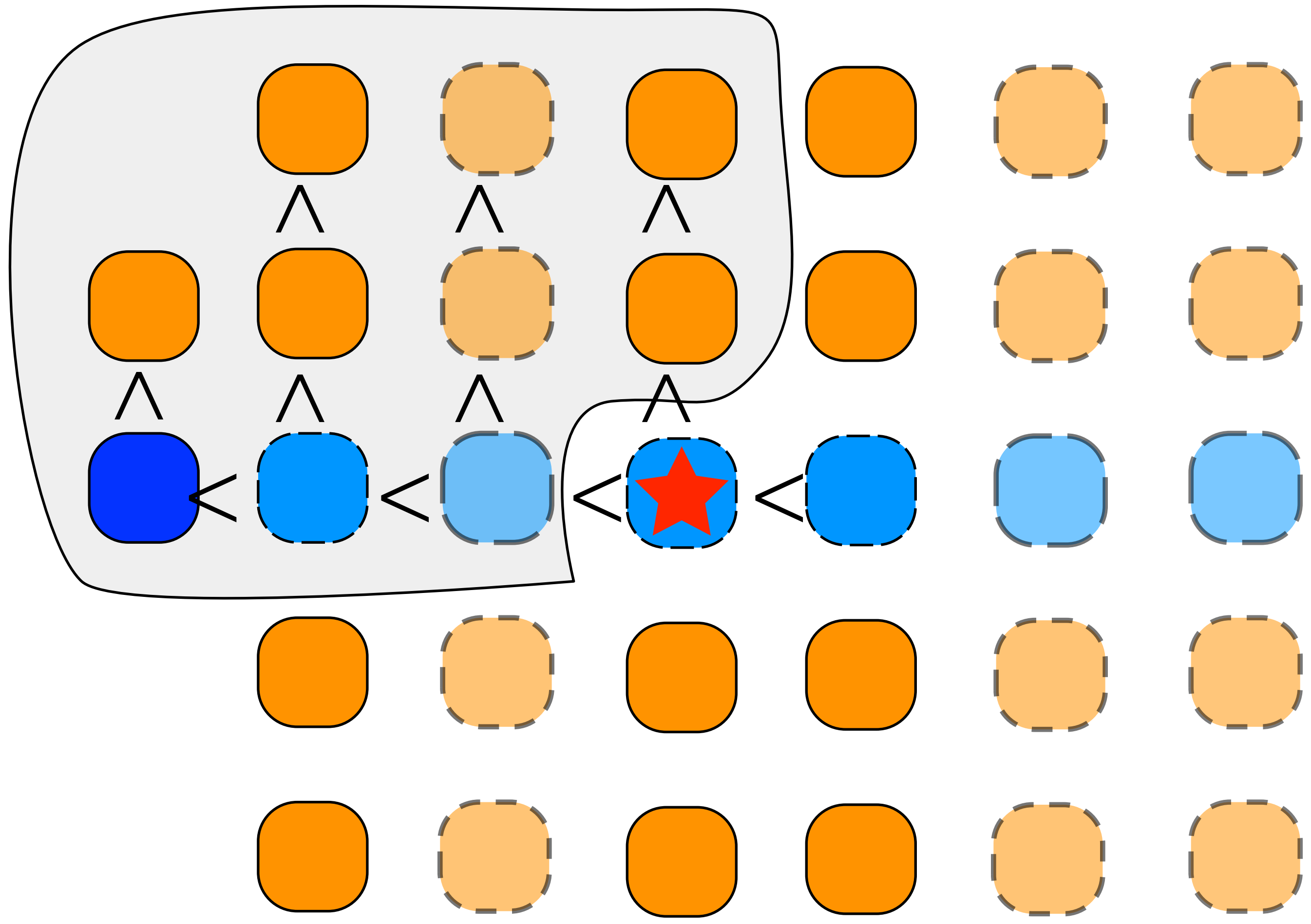
a nice property of our partition



SWITCH TO A BIGGER EXAMPLE



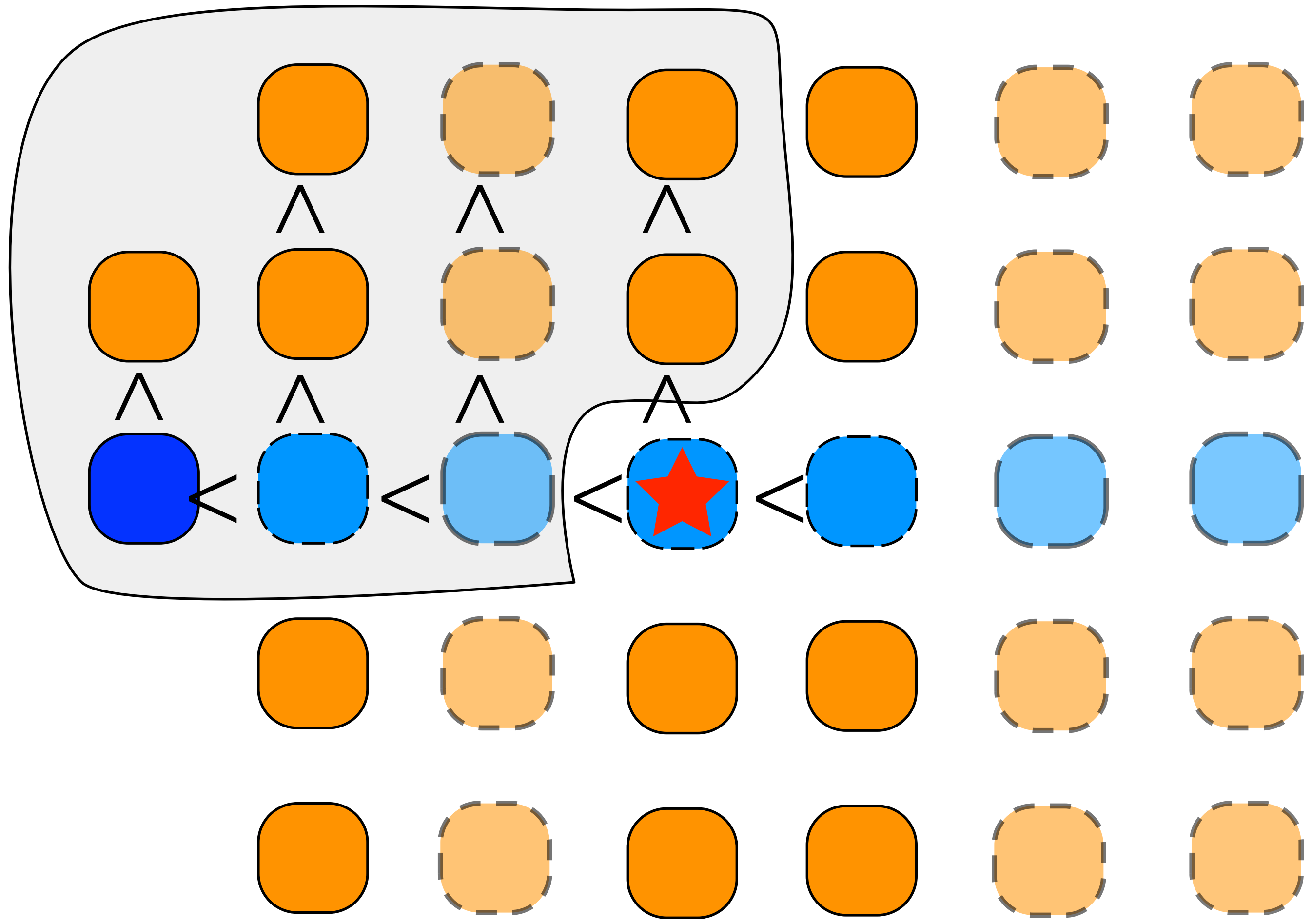
a nice property of our partition



a nice property of our partition

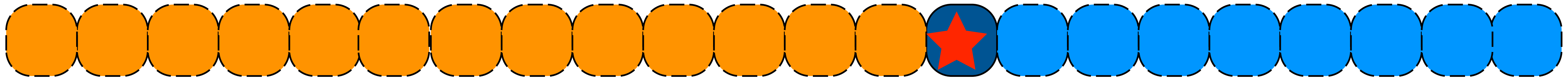
$$3\left(\left\lceil \frac{1}{2}\lceil n/5\rceil \right\rceil - 2\right)$$

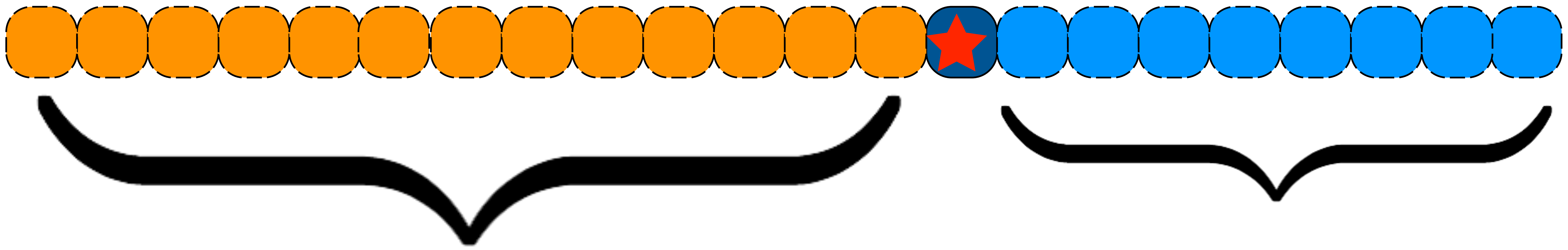
$$\geq \frac{3n}{10} - 6$$



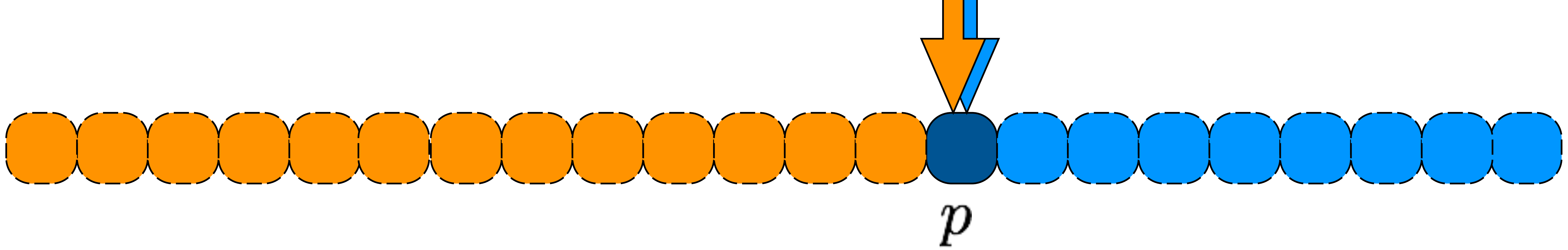
this implies there are
at most $\frac{7n}{10}+6$ numbers
larger than ★
/smaller

a nice property of our partition

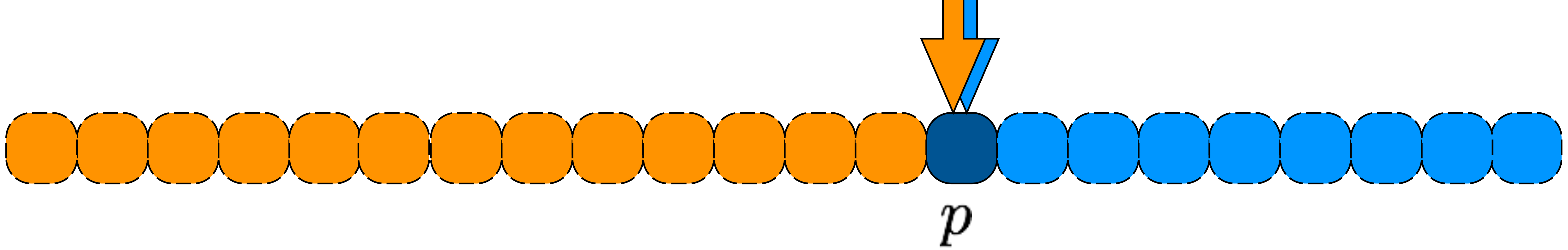




$$\le \frac{7n}{10} + 6 \qquad\qquad \le \frac{7n}{10} + 6$$



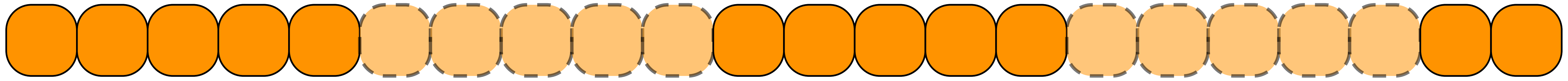
select $(i, A[1,\ldots,n])$



select $(i, A[1, \ldots, n])$

handle base case for small list
else pivot = FindPartitionValue(A,n)
partition list about pivot
if pivot is position i, return pivot
else if pivot is in position > i select $(i, A[1, \ldots, p-1])$
else select $((i-p-1), A[p+1, \ldots, n])$

FindPartition $(A[1, \ldots, n])$



divide list into groups of 5 elements

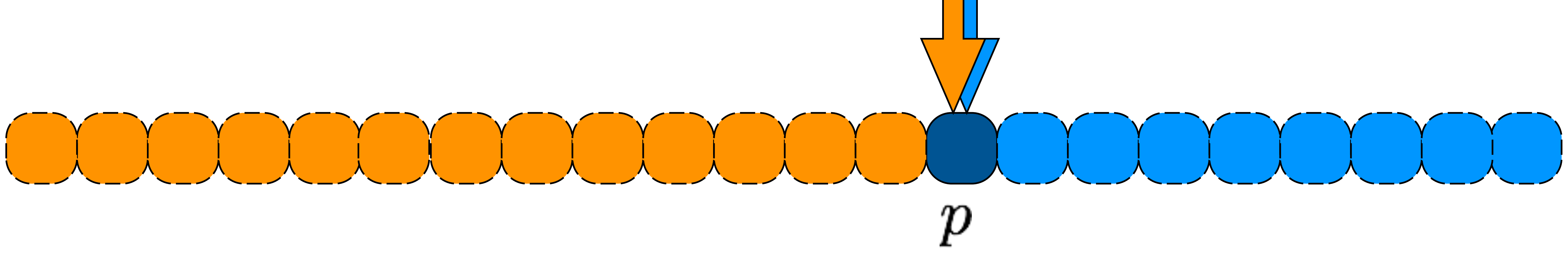
find median of each small list

gather all medians

call select(...) on this sublist to find median

return the result

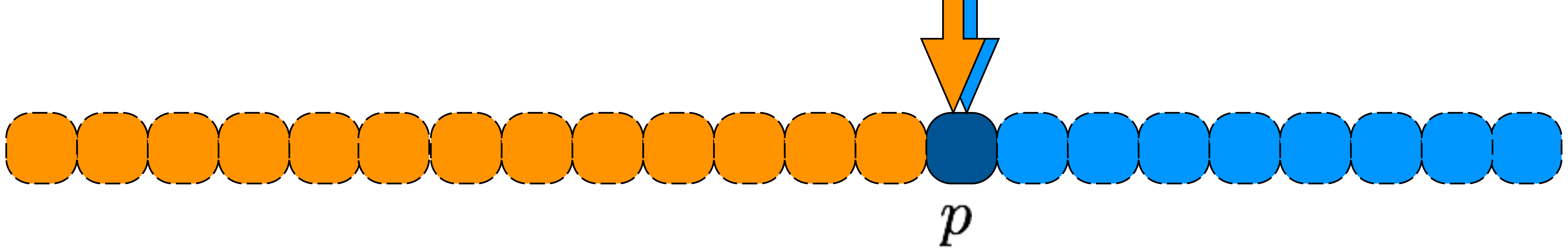
$$P(n) = S(\lceil n/5 \rceil) + O(n)$$



select $(i, A[1, \ldots, n])$

handle base case for small list
else pivot = FindPartitionValue(A,n)
partition list about pivot
if pivot is position i, return pivot
else if pivot is in position $> i$ select $(i, A[1, \ldots, p-1])$
else select $((i-p-1), A[p+1, \ldots, n])$

$$S(n) = S(\lceil n/5 \rceil) + O(n) + S(7n/10 + 6)$$



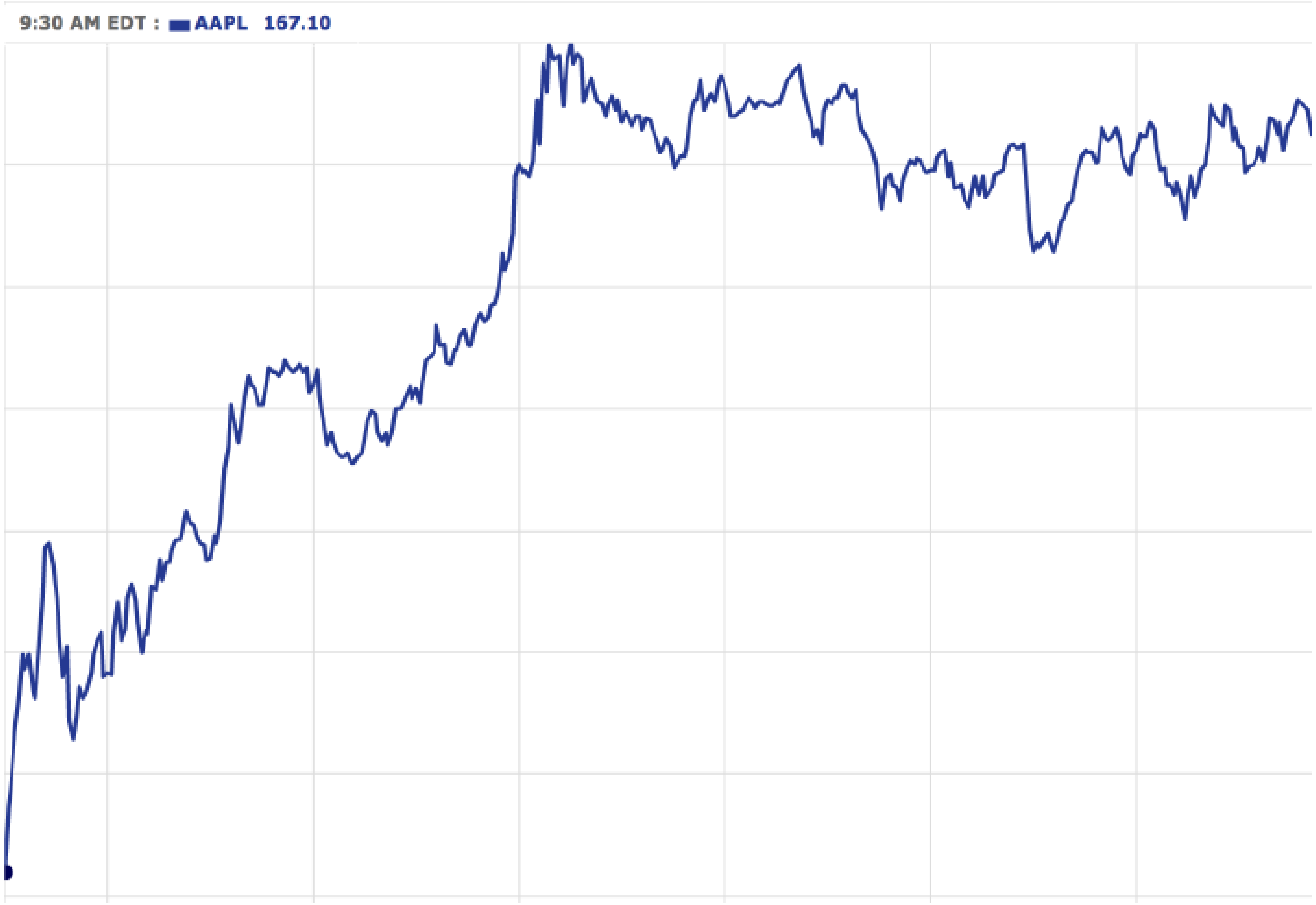
select $(i, A[1, \ldots, n])$

handle base case for small list
else pivot = FindPartitionValue(A,n)
partition list about pivot
if pivot is position i, return pivot
else if pivot is in position $> i$ select $(i, A[1, \ldots, p-1])$
else select $((i-p-1), A[p+1, \ldots, n])$

$$S(n) = S(\lceil n/5 \rceil) + O(n) + S(7n/10 + 6)$$

$$\Theta(n)$$

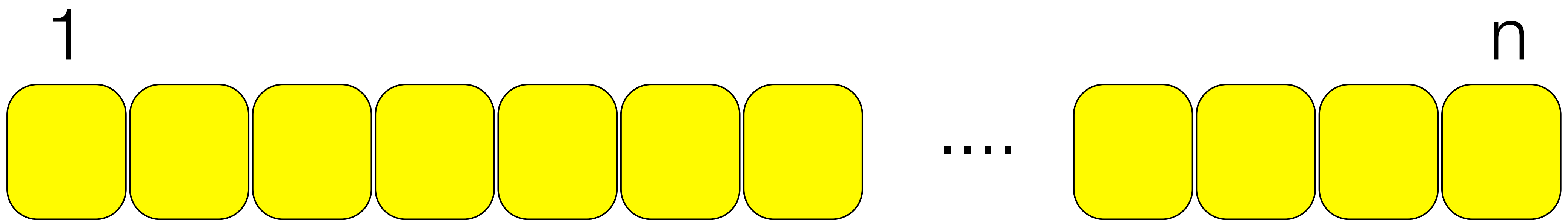
arbitrage


9:30 AM EDT : AAPL 167.10

Jan 14, 2009 :
BPT 72.59
100
90
80
70
60
© 2008 Yahoo! Inc.
2008
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
2009
Feb
Mar

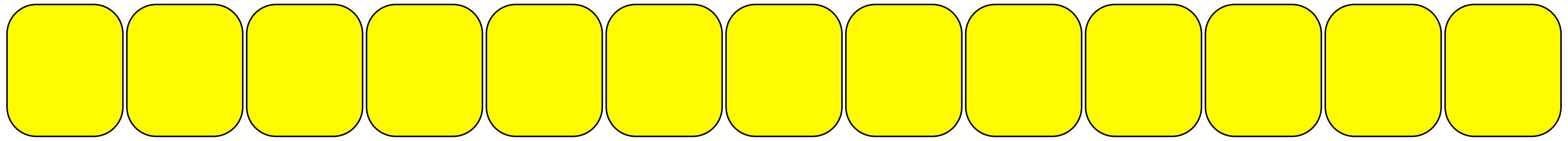
12:38 PM EDT : AIG 40.58
© 2008 Yahoo! Inc.
10:00 AM
11:00 AM
12:00 PM
1:00 PM
2:00 PM
3:00 PM

input: array of n numbers

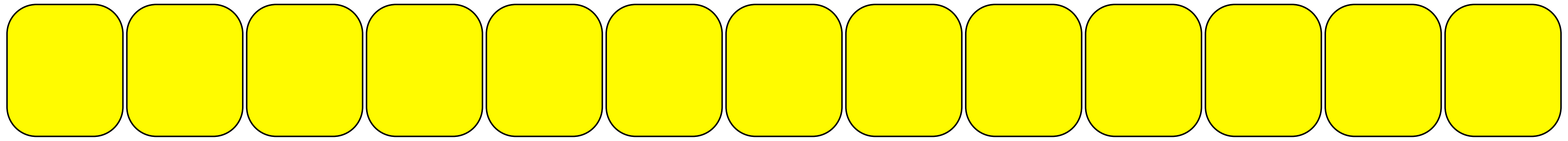


goal:

first attempt

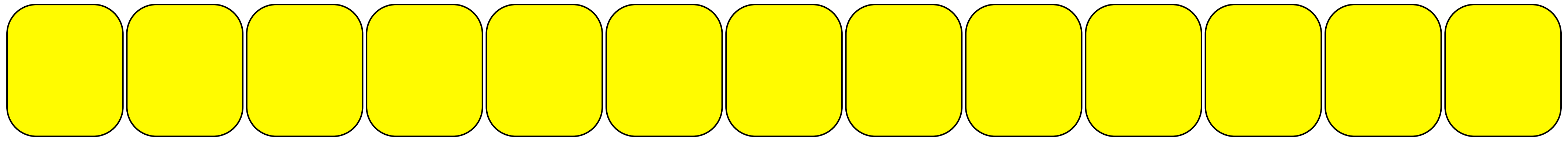


first attempt



```
arbit(A[1...n])
```

first attempt



```
arbit(A[1...n])
    base case if |A|=1
    lg = arbit(left(A))
    rg = arbit(right(A))
    minl = min(left(A))
    maxr = max(right(A))
    return max{maxr-minl,lg,rg}
```

better approach

second attempt

```
arbit+(A[1...n])
  base case if |A|=1
```

second attempt

```
arbit+(A[1...n])
  base case if |A|=1
   (lg,minl,max) = arbit(left(A))
   (rg,mi,maxr) = arbit(right(A))

   return max{maxr-minl,lg,rg}
```