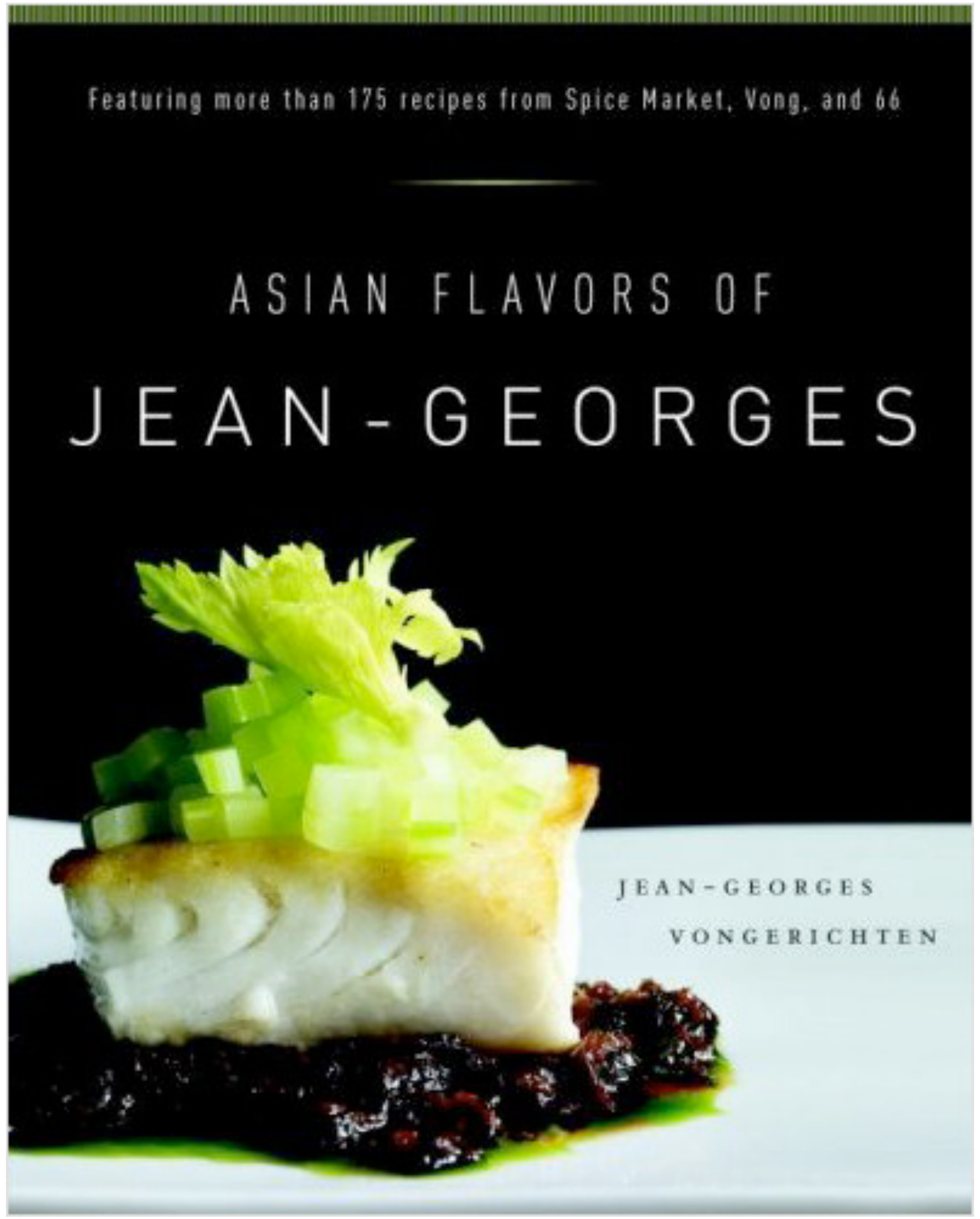
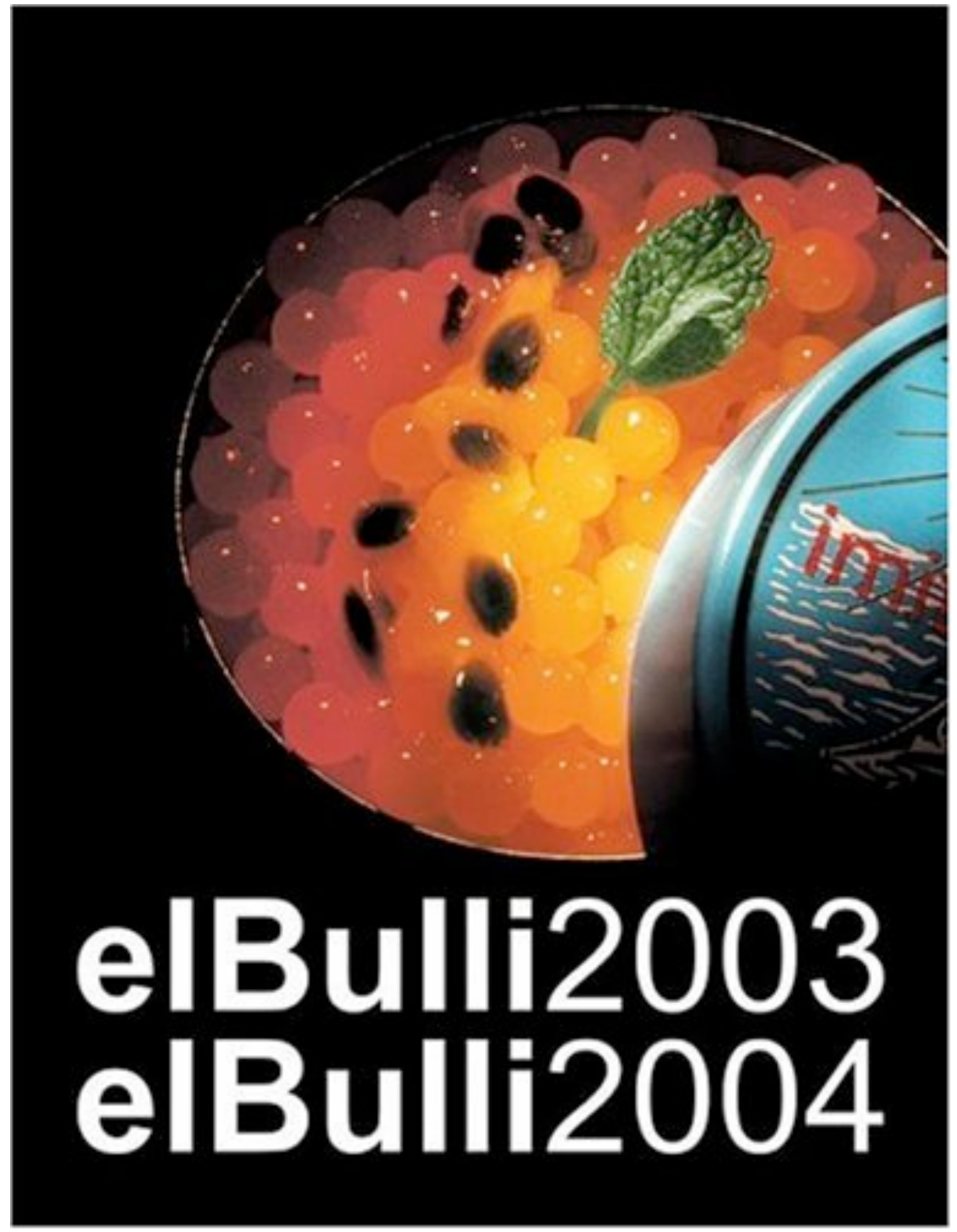
L5

feb 4 2016

4102
shelat

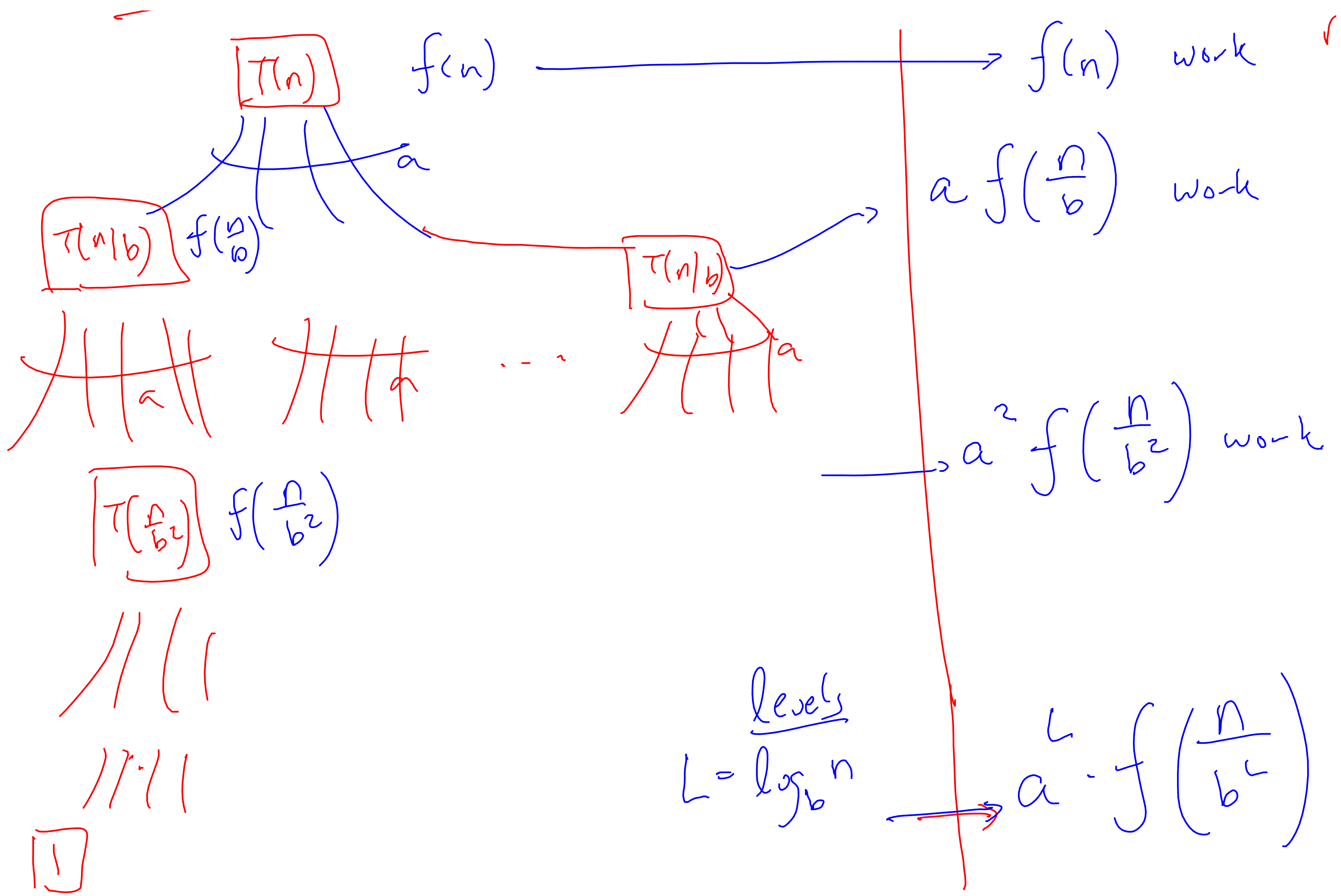
cookbook


Featuring more than 175 recipes from Spice Market, Vong, and 66
ASIAN FLAVORS OF
JEAN-GEORGES
JEAN-GEORGES
VONGERICHTEN


elBulli2003
elBulli2004

$$T(n) = aT(n/b) + f(n)$$

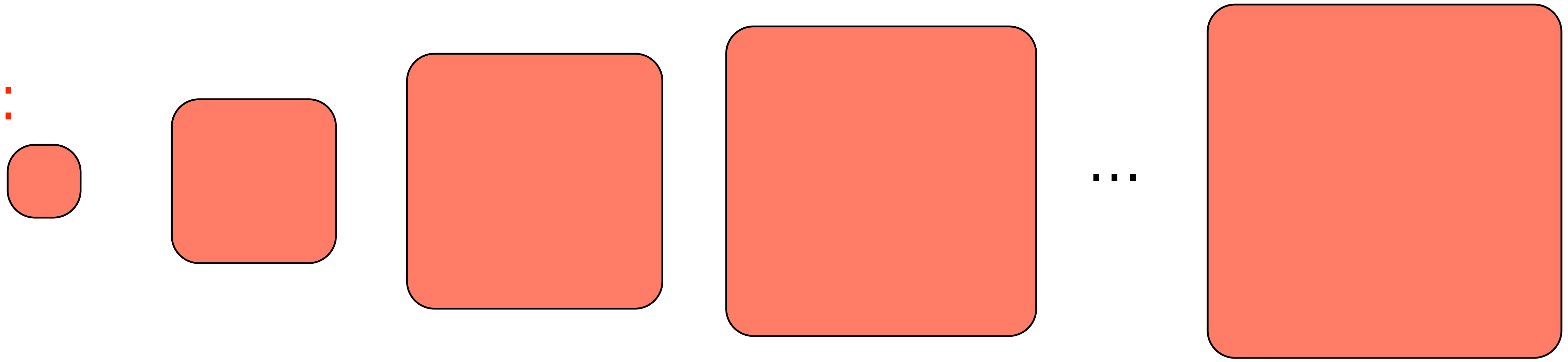
$$T(n) = aT(n/b) + f(n)$$


T(n)
f(n)
a
T(n/b)
f(n/b)
T(n/b)
a
a
a
T(n/b^2)
f(n/b^2)
1
f(n) work
a f(n/b) work
a^2 f(n/b^2) work
levels
L = log_b n
a^L · f(n/b^L)

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

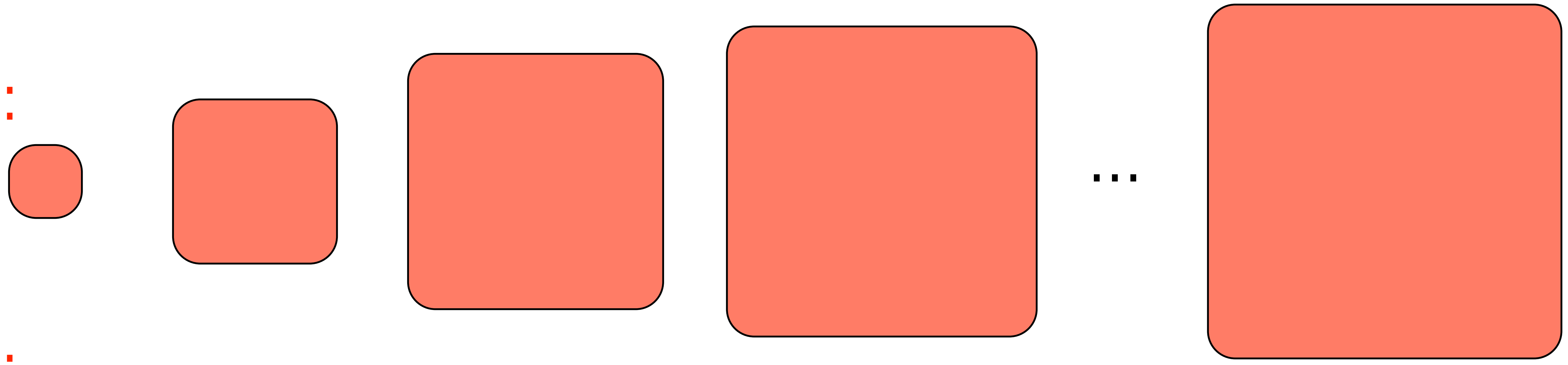
$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 1:

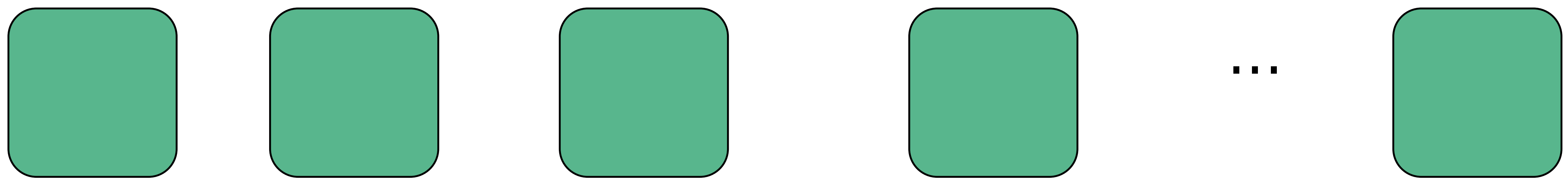


$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 1:

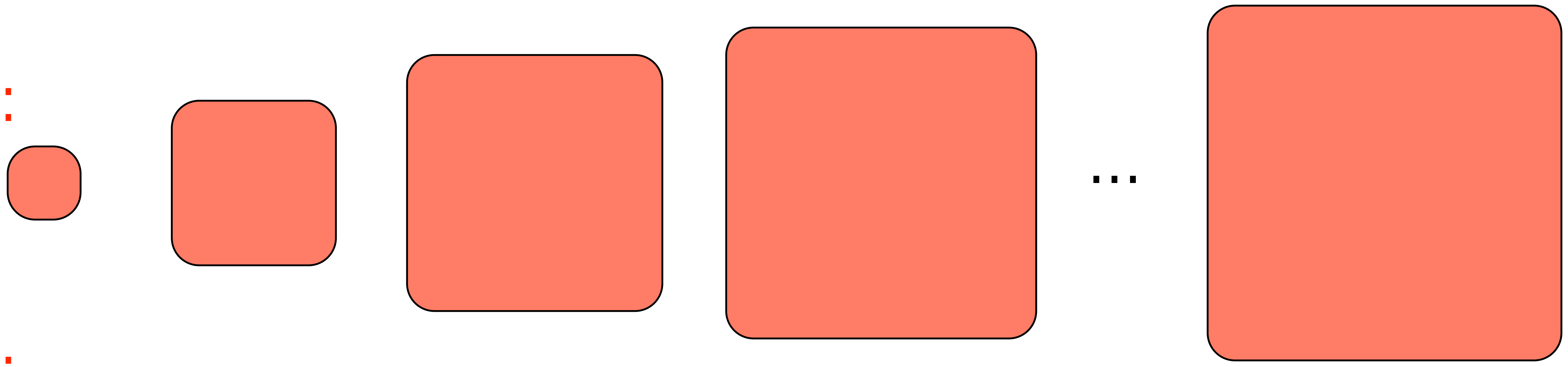


case 2:

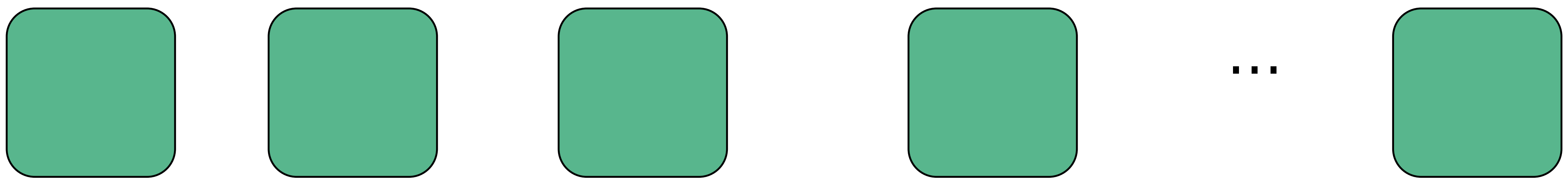


$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

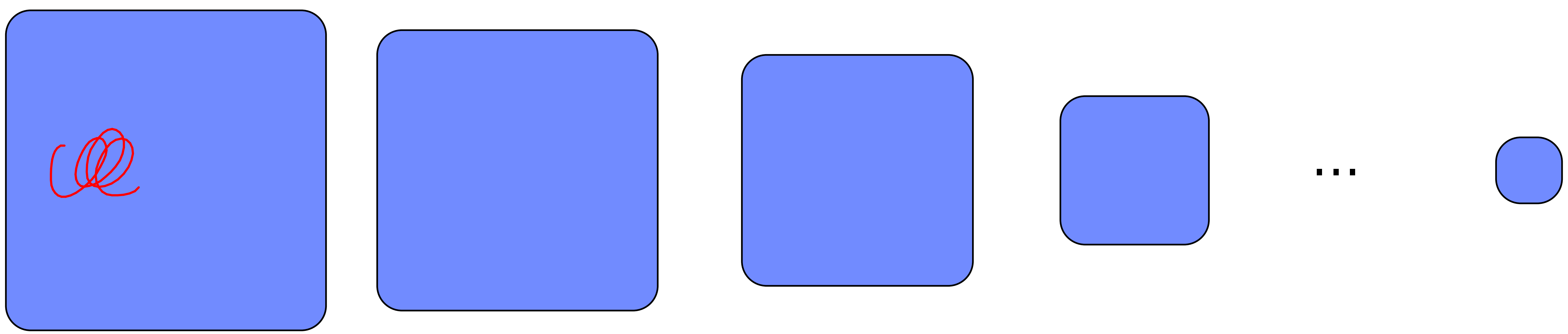
case 1:



case 2:

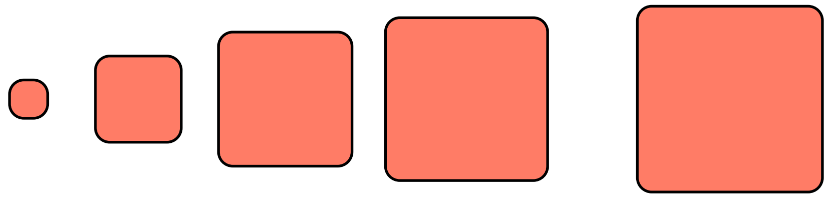


case 3:

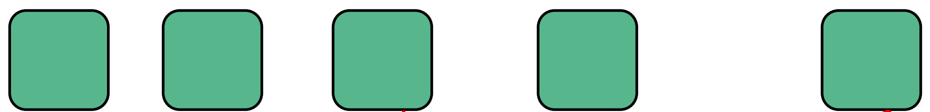


$$T(n) = aT(n/b) + f(n)$$

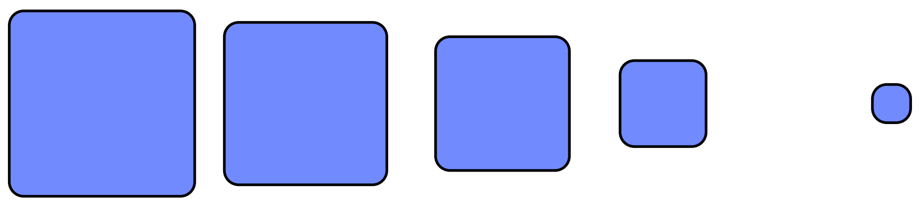
Masters

case 1: 

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

case 2: 

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

case 3: 

$$f(n) = \Omega(n^{\log_b a + \epsilon}), \epsilon > 0$$

and c<1 s.t $af(n/b) < cf(n)$

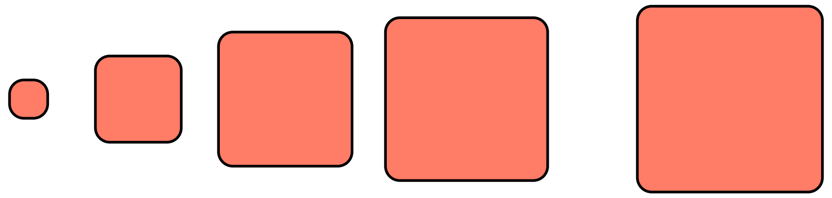
Then:

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

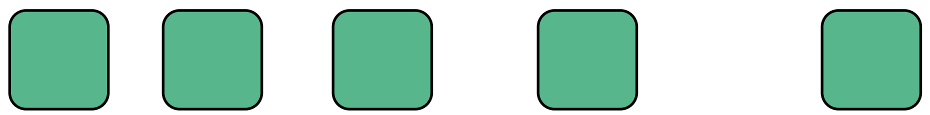
$$T(n) = \Theta(n^{\log_b a} \cdot \log(n))$$

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

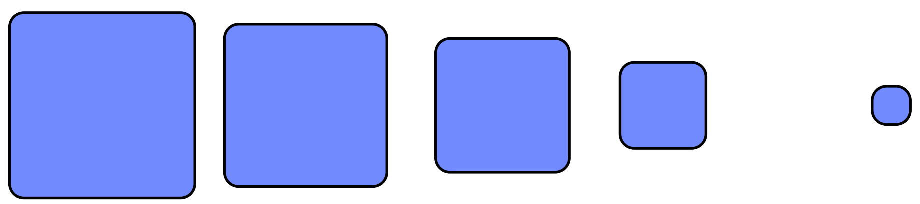
$T(n) = aT(n/b) + f(n)$

case 1: 

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

case 2: 

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

case 3: 

$$f(n) = \Omega(n^{\log_b a + \epsilon}), \epsilon > 0$$

and c<1 s.t $af(n/b) < cf(n)$

Then:

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

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

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

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 1: Since $f(n) < cn^{\log_b a - \epsilon}$

We have: $T(n) < cn^{\log_b a - \epsilon} + a\left(\frac{n}{b}\right)^{\log_b a - \epsilon} + \cdots + \cdots$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 1: Since $f(n) < cn^{\log_b a - \epsilon}$

We have:

$$T(n) \le cn^{\log_b a - \epsilon}\left[1 + \frac{a}{b^{\log_b a - \epsilon}} + \frac{a^2}{(b^2)^{\log_b a - \epsilon}} + \frac{a^{L-1}}{(b^{L-1})^{\log_b a - \epsilon}}\right] + n^{\log_b a}\, c$$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 1: Since $f(n) < cn^{\log_b a - \epsilon}$

We have:

$$T(n) \leq cn^{\log_b a-\epsilon}\left[1 + \frac{a}{b^{\log_b a-\epsilon}} + \frac{a^2}{(b^2)^{\log_b a-\epsilon}} + \frac{a^{L-1}}{(b^{L-1})^{\log_b a-\epsilon}}\right] + n^{\log_b a}\, c$$

$$T(n) \leq cn^{\log_b a-\epsilon}\left[1 + b^{\epsilon} + b^{2\epsilon} + \cdots + b^{\epsilon(L-1)}\right] + n^{\log_b a}\, c$$

$$1 + b^{\epsilon} + b^{2\epsilon} + \cdots + b^{(L-1)\epsilon} = \left(\frac{b^{\epsilon L} - 1}{b^{\epsilon} - 1}\right) = \left(\frac{n^{\epsilon} - 1}{b^{\epsilon} - 1}\right)$$

① $$\frac{a^i}{(b^i)^{\log_b(a) - \epsilon}} = \frac{a^i}{(b^{\log_b(a) - \epsilon})^i} = \frac{a^i}{\frac{a^i}{b^{\epsilon i}}} = b^{\epsilon i}$$

② $$a^L = a^{\log_b n} = (b^{\log_b a})^{\log_b n} = (b^{\log_b n})^{\log_b a} = n^{\log_b a}$$

③ $$b^{\epsilon L} = (b^{\log_b n})^{\epsilon} = n^{\epsilon}$$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 1: Since $f(n) < cn^{\log_b a - \epsilon}$

We have:

$$T(n) \leq cn^{\log_b a-\epsilon}\left[1 + \frac{a}{b^{\log_b a-\epsilon}} + \frac{a^2}{(b^2)^{\log_b a-\epsilon}} + \frac{a^{L-1}}{(b^{L-1})^{\log_b a-\epsilon}}\right] + n^{\log_b a}\, c$$

$$T(n) \leq cn^{\log_b a-\epsilon}\left[1 + b^{\epsilon} + b^{2\epsilon} + \cdots + b^{\epsilon(L-1)}\right] + n^{\log_b a}\, c$$

$$T(n) \leq cn^{\log_b a-\epsilon}\left[\frac{b^{\epsilon L} - 1}{b^{\epsilon} - 1}\right] + n^{\log_b a}\, c$$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 1: Since $f(n) < cn^{\log_b a - \epsilon}$

We have:

$$T(n) \leq cn^{\log_b a - \epsilon}\left[1 + \frac{a}{b^{\log_b a - \epsilon}} + \frac{a^2}{(b^2)^{\log_b a - \epsilon}} + \frac{a^{L-1}}{(b^{L-1})^{\log_b a - \epsilon}}\right] + n^{\log_b a}\, c$$

$$T(n) \leq cn^{\log_b a - \epsilon}\left[1 + b^{\epsilon} + b^{2\epsilon} + \cdots + b^{\epsilon(L-1)}\right] + n^{\log_b a}\, c$$

$$T(n) \leq cn^{\log_b a - \epsilon}\left[\frac{b^{\epsilon L} - 1}{b^{\epsilon} - 1}\right] + n^{\log_b a}\, c$$

$$T(n) \leq c' n^{\log_b a - \epsilon}\left[n^{\epsilon} - 1\right] + n^{\log_b a} c \quad \Rightarrow \quad T(n) = O\left(n^{\log_b a}\right)$$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 1: Since $f(n) < cn^{\log_b a - \epsilon}$

We have:

$$T(n) \leq cn^{\log_b a - \epsilon}\left[1 + \frac{a}{b^{\log_b a - \epsilon}} + \frac{a^2}{(b^2)^{\log_b a - \epsilon}} + \frac{a^{L-1}}{(b^{L-1})^{\log_b a - \epsilon}}\right] + n^{\log_b a}\, c$$

$$T(n) \leq cn^{\log_b a - \epsilon}\left[1 + b^{\epsilon} + b^{2\epsilon} + \cdots + b^{\epsilon(L-1)}\right] + n^{\log_b a}\, c$$

$$T(n) \leq cn^{\log_b a - \epsilon}\left[\frac{b^{\epsilon L} - 1}{b^{\epsilon} - 1}\right] + n^{\log_b a}\, c$$

$$T(n) \leq c' n^{\log_b a - \epsilon}\left[n^{\epsilon} - 1\right] + n^{\log_b a} c = O(n^{\log_b a})$$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right) > 1$$

case 1: Lower bound

We have: $T(n) \geq a^L \cdot c = n^{\log_b(a)} \cdot c$

$$\Rightarrow T(n) = \Omega\left(n^{\log_b(a)}\right)$$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 1: Lower bound

We have:

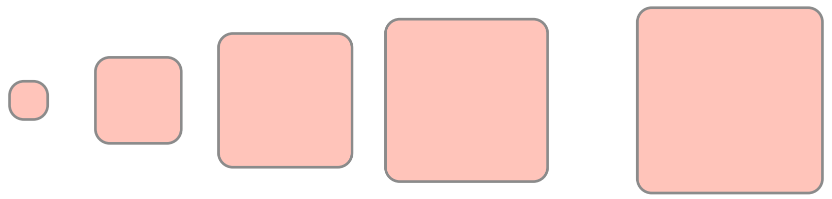
$$T(n) \geq a^L f(\frac{n}{b^L})$$

$$\geq a^{\log_b(n)} = (b^{\log_b a})^{\log_b(n)}$$

$$= n^{\log_b(a)}$$

$$= \Omega(n^{\log_b(a)})$$

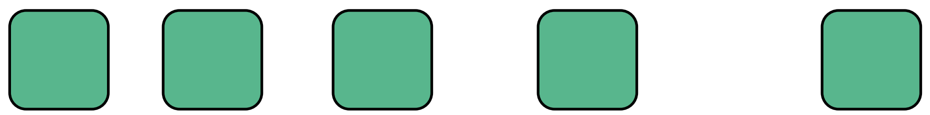
$T(n) = aT(n/b) + f(n)$

case 1: 

Then:

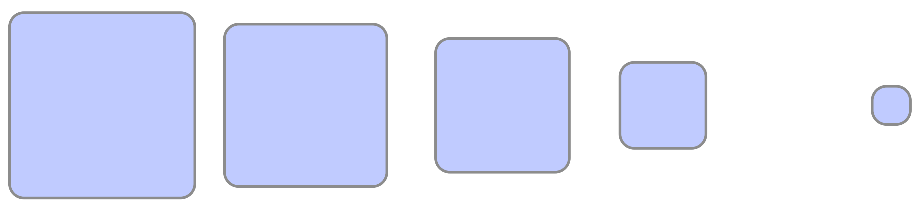
$f(n) = O(n^{\log_b a - \epsilon}), \epsilon > 0$

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

case 2: 

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

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

case 3: 

$f(n) = \Omega(n^{\log_b a + \epsilon}), \epsilon > 0$

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

and c<1 s.t $af(n/b) < cf(n)$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 2 : When $f(n) < cn^{\log_b a}$

$T(n) \le c \cdot n^{\log_b a} + a \cdot c\left(\frac{n}{b}\right)^{\log_b a} + a^2 \cdot c\left(\frac{n}{b^2}\right)^{\log_b a} + \cdots + a^{L-1} \cdot c\left(\frac{n}{b^{L-1}}\right)^{\log} + a^{L}$ ---

$= cn^{\log_b a}\left[\frac{a}{a} + \frac{a^2}{a^2} + \cdots + \frac{a^{L-1}}{a^{L-1}}\right] + a^L \cdot \left(\frac{n}{b^L}\right)^{\log_b a}$

$= cn^{\log_b a}[\underbrace{1 + 1 + \cdots + 1 + 1}_{L \sim \log_b(n)}]$

$= cn^{\log_b a} \cdot \log_b(n) + \cdots = O(n^{\log_b a} \cdot \log n)$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 2 : When $f(n) < cn^{\log_b a}$

$$T(n) \le cn^{\log_b(a)}\left[1 + \frac{a}{b^{\log_b(a)}} + \frac{a^2}{(b^2)^{\log_b(a)}} + \cdots + \frac{a^{L-1}}{(b^{L-1})^{\log_b(a)}}\right] + cn^{\log_b(a)}$$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 2 : When $f(n) < cn^{\log_b a}$

$$T(n) \leq cn^{\log_b(a)} \left[1 + \frac{a}{b^{\log_b(a)}} + \frac{a^2}{(b^2)^{\log_b(a)}} + \cdots + \frac{a^{L-1}}{(b^{L-1})^{\log_b(a)}}\right] + cn^{\log_b(a)}$$

$$T(n) \leq cn^{\log_b(a)} [1 + 1 + \cdots + 1] + cn^{\log_b(a)}$$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 2 : When $f(n) < cn^{\log_b a}$

$$T(n) \leq cn^{\log_b(a)}\left[1 + \frac{a}{b^{\log_b(a)}} + \frac{a^2}{(b^2)^{\log_b(a)}} + \cdots + \frac{a^{L-1}}{(b^{L-1})^{\log_b(a)}}\right] + cn^{\log_b(a)}$$

$$T(n) \leq cn^{\log_b(a)}\left[1 + 1 + \cdots + 1\right] + cn^{\log_b(a)}$$

$$\leq cn^{\log_b(a)}\left[\log_b(n)\right]$$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 2 : When $f(n) < cn^{\log_b a}$

$$T(n) \le cn^{\log_b(a)}\left[1 + \frac{a}{b^{\log_b(a)}} + \frac{a^2}{(b^2)^{\log_b(a)}} + \cdots + \frac{a^{L-1}}{(b^{L-1})^{\log_b(a)}}\right] + cn^{\log_b(a)}$$

$$T(n) \le cn^{\log_b(a)}\left[1 + 1 + \cdots + 1\right] + cn^{\log_b(a)}$$

$$\le cn^{\log_b(a)}\left[\log_b(n)\right]$$

$$= O(n^{\log_b a} \log n)$$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

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

case 2 : When $f(n) > cn^{\log_b(a)}$

lower-bound

$$T(n) \geq cn^{\log_b(a)} \left[1 + \frac{a}{b^{\log_b(a)}} + \frac{a^2}{(b^2)^{\log_b(a)}} + \cdots + \frac{a^{L-1}}{(b^{L-1})^{\log_b(a)}}\right]$$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 2 : When $f(n) > cn^{\log_b(a)}$

lower-bound

$$T(n) \geq cn^{\log_b(a)}\left[1 + \frac{a}{b^{\log_b(a)}} + \frac{a^2}{(b^2)^{\log_b(a)}} + \cdots + \frac{a^{L-1}}{(b^{L-1})^{\log_b(a)}}\right]$$

$$T(n) \geq cn^{\log_b(a)}\left[1 + 1 + \cdots + 1\right]$$

$$\geq cn^{\log_b(a)}\log_b(a)$$

$$\Omega(n^{\log_b(a)}\log_b(a))$$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

$f(n) = \Omega(n^{\log_b a + \epsilon})$

case 3: $f(n) > n^{\log_b a + \epsilon}$ and c<1 s.t $af(n/b) < cf(n)$

$af\left(\frac{n}{b}\right) < c \cdot f(n)$

$a^2 f\left(\frac{n}{b^2}\right) = a \cdot \left(a \cdot f\left(\frac{n/b}{b}\right)\right) < a\left(c \cdot f\left(\frac{n}{b}\right)\right) \leq c^2 f(n)$

$< c \cdot f\left(\frac{n}{b}\right)$

$a^3 f\left(\frac{n}{b^3}\right) \leq c^3 f(n) \quad \ldots$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 3: $f(n) > n^{\log_b a + \epsilon}$ and c<1 s.t $af(n/b) < cf(n)$

$$T(n) \leq f(n) + cf(n) + c^2 f(n) + \cdots + c^L f(n)$$

$$f(n)\left[1 + c + c^2 + \cdots c^L\right] = O(f(n))$$

because c is a constant c < 1

$$2^L = 2^{\log n} = n$$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 3: $f(n) > n^{\log_b a + \epsilon}$ and c<1 s.t $af(n/b) < cf(n)$

$$T(n) \leq f(n) + cf(n) + c^2 f(n) + \cdots + c^L f(n)$$

$$T(n) \leq f(n)[1 + c + c^2 + \cdots + c^L]$$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

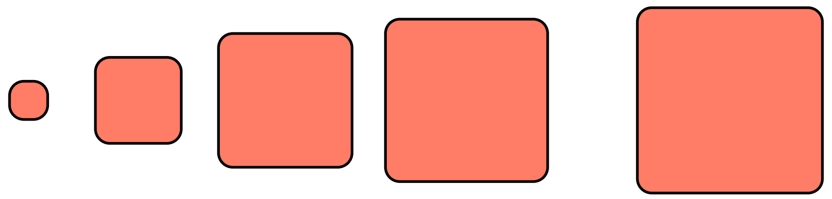
case 3: $f(n) > n^{\log_b a + \epsilon}$ and c<1 s.t $af(n/b) < cf(n)$

$$T(n) \leq f(n) + cf(n) + c^2 f(n) + \cdots + c^L f(n)$$

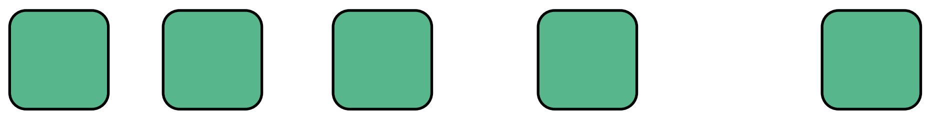
$$T(n) \leq f(n)[1 + c + c^2 + \cdots + c^L]$$

$$= O(f(n))$$

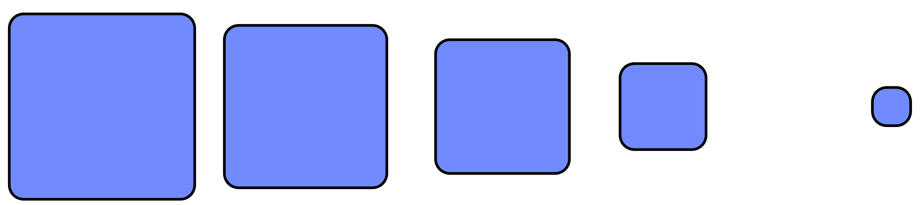
$$T(n) = aT(n/b) + f(n)$$

case 1: 

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

case 2: 

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

case 3: 

$$f(n) = \Omega(n^{\log_b a + \epsilon}), \epsilon > 0$$

and c<1 s.t $af(n/b) < cf(n)$

Then:

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

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

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

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

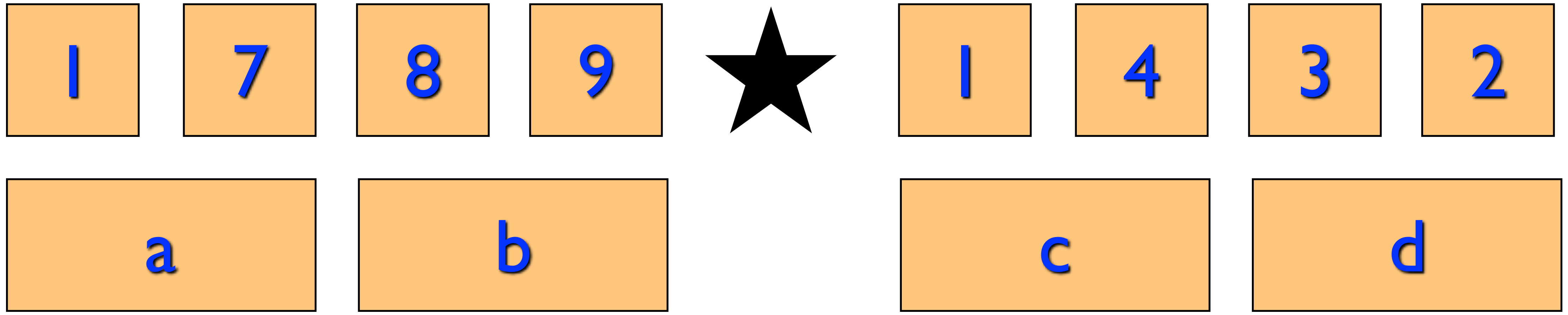
a — b — f(n)

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

$n^{\log_2 8} = n^3$

$\Theta(n^2) = O(n^{3-\epsilon})$ for $\epsilon = 0.01$, by

Case I, $T(n) = \Theta(n^3)$



$$T(n) = 4T(n/2) + 3O(n)$$

example 2:

$$T(n) = T\left(\frac{14}{17}n\right) + 24$$

$n^{0-\epsilon}$

$f(n) = \Theta(1)$　　$a = 1$　　$b = \frac{17}{14}$　　$n^{\log_{\frac{17}{14}} 1} = n^0 = \Theta(1)$

case I (No)

case 2 (yes) — so $T(n) = \Theta(n^0 \cdot \log n)$

$= \Theta(\log n)$

$$T(n) = \underbrace{2}_{a}T(n/\underbrace{2}_{b}) + n^3$$

$f(n) = n^3$ $\quad\quad n^{\log_2 2} = n = \Theta(n^1)$

$f(n) = \Omega(n^{1+\epsilon})$ for eps > 0.

is there a c s.t. $2 \cdot f(\frac{n}{2}) < c \cdot f(n)$

$$2\left(\frac{n}{2}\right)^3 = 2\frac{n^3}{8} = \frac{n^3}{4} < \frac{1}{3} \cdot (n)^3$$

So if $c = \frac{1}{3}$, condition holds. $\Rightarrow T(n) = \Theta(n^3)$

$$T(n) = 16T(n/4) + n^2$$

case 2

$f(n) = n^2$

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

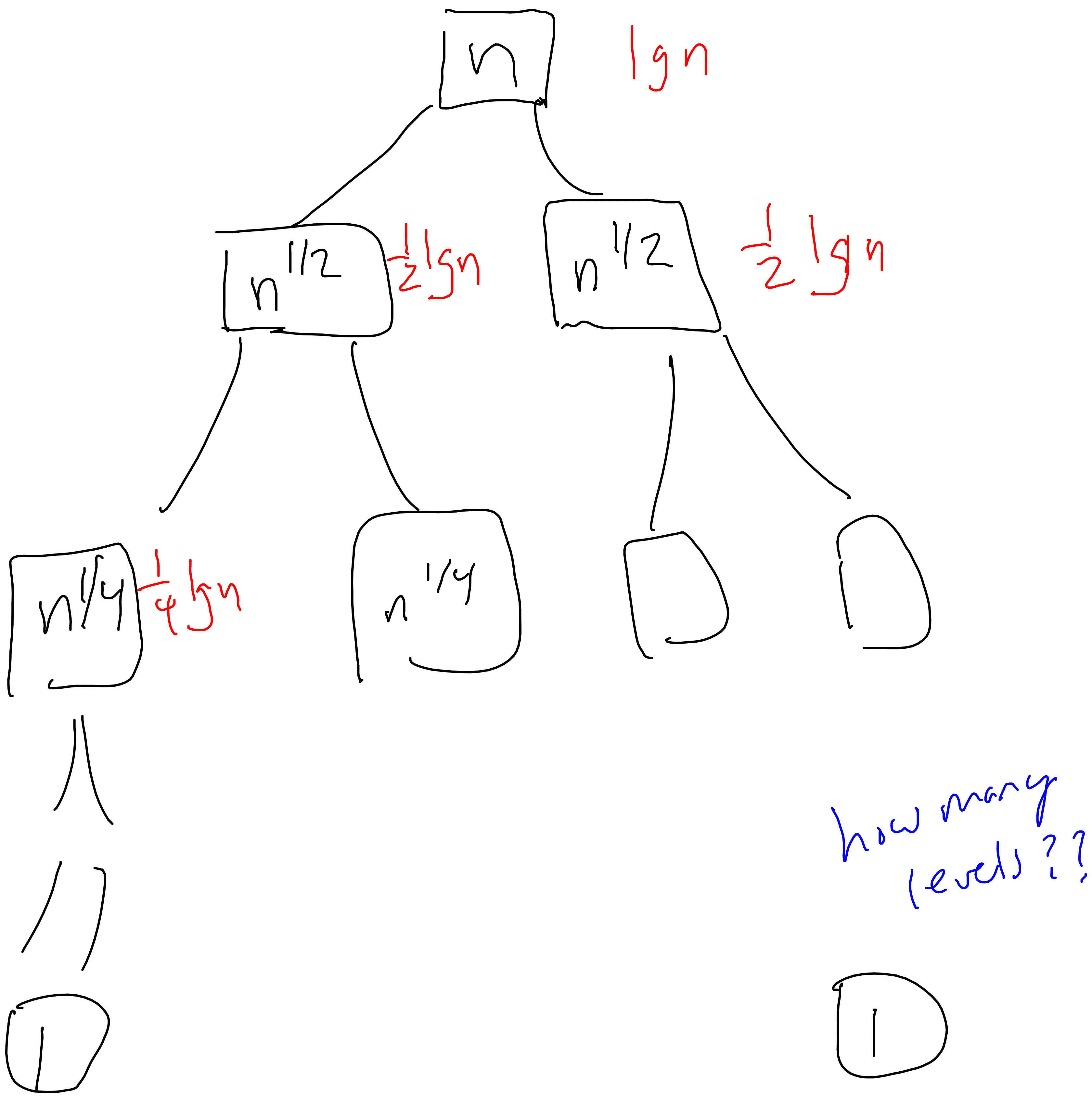
$$T(n) = 7T(n/2) + O(n^2) \Rightarrow$$

case 1

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


8
13
FIFA

$$T(n) = 2T(\sqrt{n}) + \lg n$$



$\to \lg n$

$\to \lg n$

$\to \lg n$

$\to \lg n$

good guess:

$$T(n) = O(\lg n \cdot \log\log n)$$

$$n^{\frac{1}{2^L}} = 2$$

$$\left(\frac{1}{2^L}\right)\log(n) = 1$$

$$\Rightarrow \log n = 2^L \Rightarrow L = \log\log(n)$$

$$T(n) = 2T(\sqrt{n}) + \lg n$$

① $2^m = n$

$m = \log n$

$T(2^m) = 2T(2^{m/2}) + c \cdot m$

② $S(m) = T(2^m)$

by definition

$S(m) = 2S(m/2) + c \cdot m$

case 2. of Masters thm.

$S(m) = \Theta(m \cdot \log m)$

$\Rightarrow$

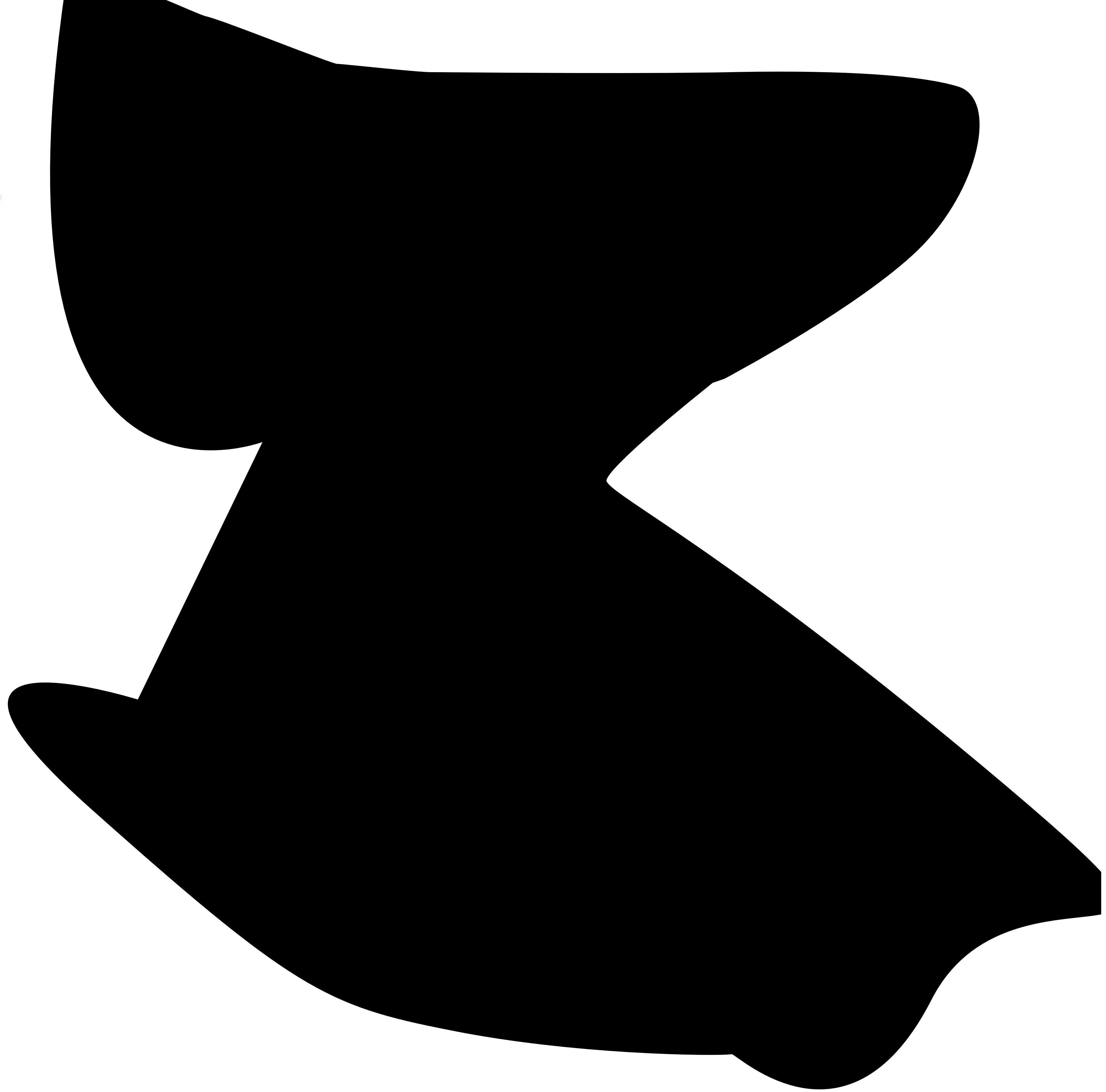
$\Rightarrow T(n) = \Theta(\log n \cdot \log \log n)$

$$\log(\log_{10} n) = \log(d \cdot \log_2 n)$$

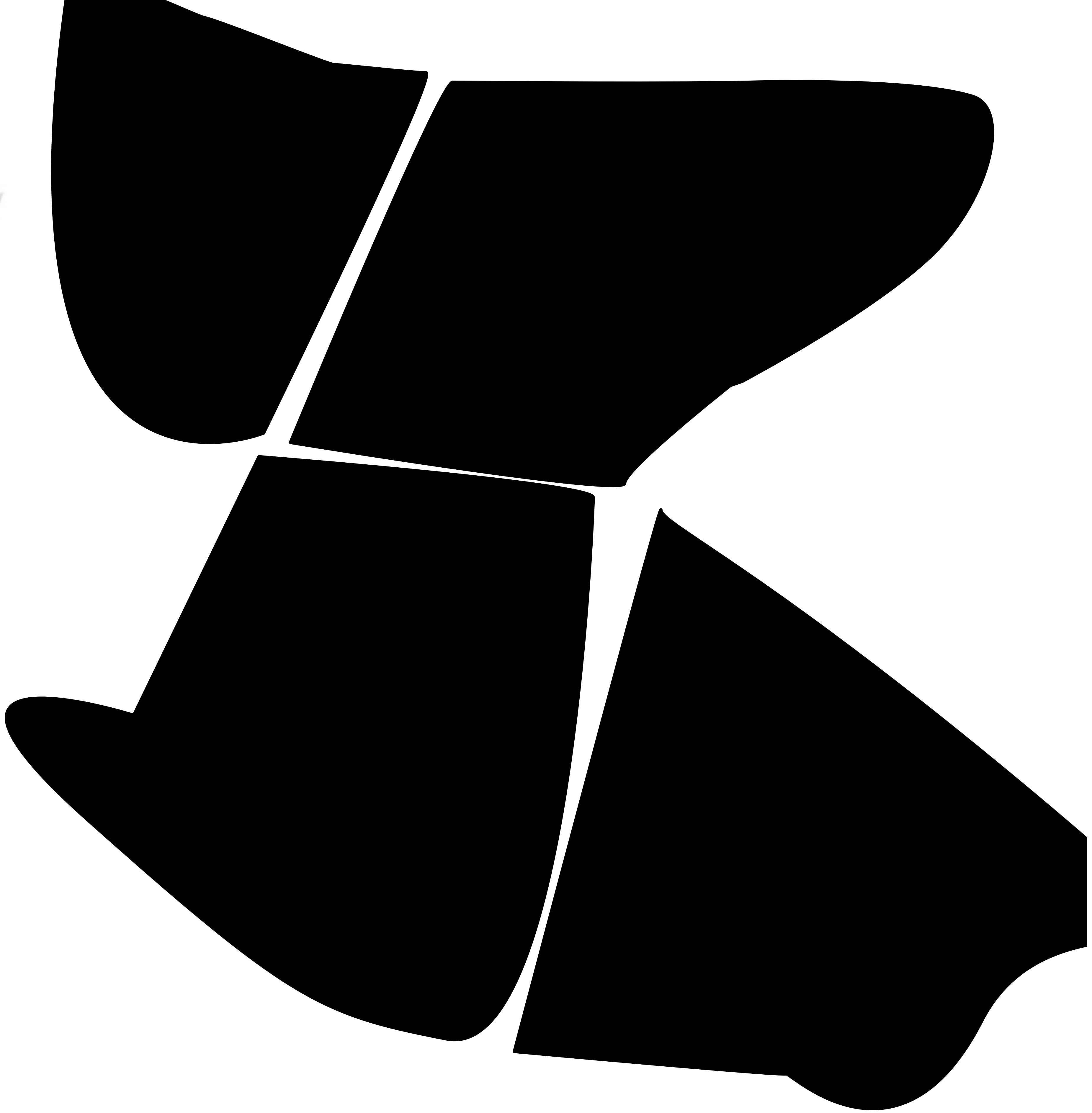
$$= \underline{\log(d)} + \underline{\underline{\log \log(n)}}$$

divide

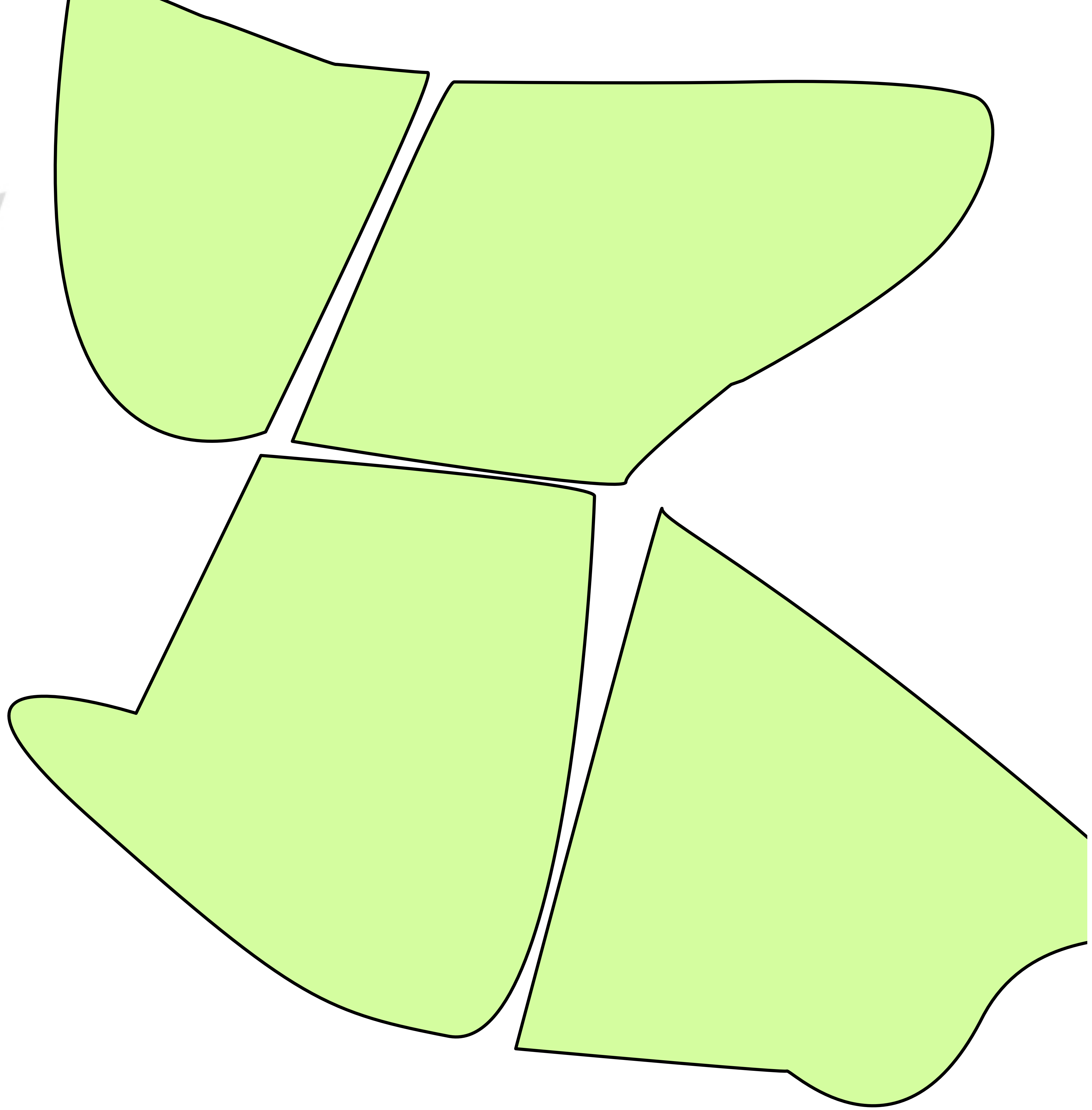
& conquer



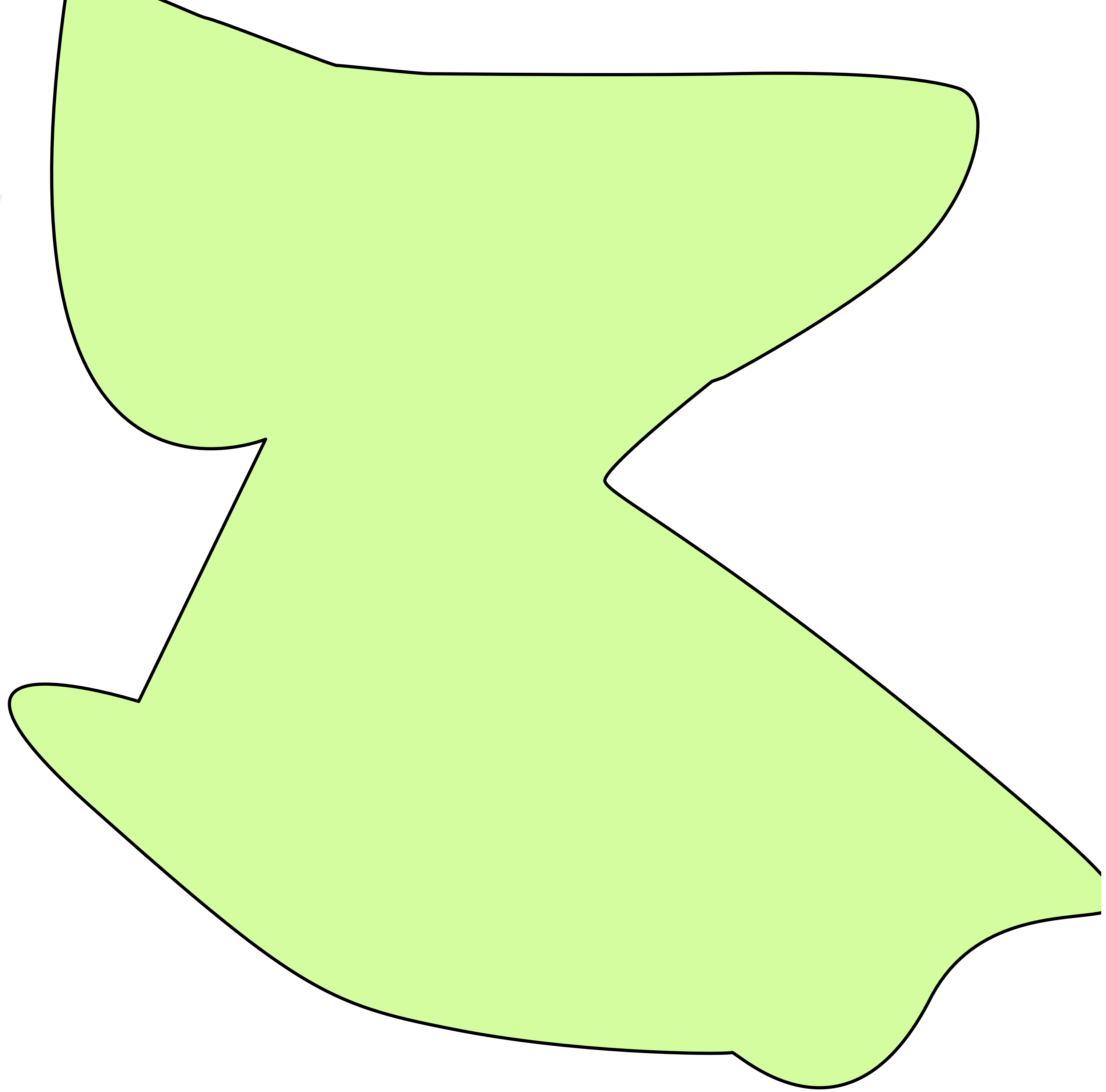
http://www.kitchenknifedrawer.com/files/1696205/uploaded/K6615D.jpg



http://www.kitchenknifedrawer.com/files/1696205/uploaded/K6615D.jpg



http://www.kitchenknifedrawer.com/files/1696205/uploaded/K6615D.jpg

http://www.kitchenknifedrawer.com/files/1696205/uploaded/K6615D.jpg

examples

mergesort

Karatsuba

- closest pair

- arbitrage

- FFT

Merge

merge-sort (A, p, r)
    if $p < r$
        $q \leftarrow \lfloor (p+r)/2 \rfloor$
        merge-sort (A, p, q)
        merge-sort $(A, q+1, r)$
        merge(A, p, q, r)

MERGE$(A[1..n], m)$:
    $i \leftarrow 1;\ j \leftarrow m+1$
    for $k \leftarrow 1$ to n
        if $j > n$
            $B[k] \leftarrow A[i];\ i \leftarrow i+1$
        else if $i > m$
            $B[k] \leftarrow A[j];\ j \leftarrow j+1$
        else if $A[i] < A[j]$
            $B[k] \leftarrow A[i];\ i \leftarrow i+1$
        else
            $B[k] \leftarrow A[j];\ j \leftarrow j+1$
    for $k \leftarrow 1$ to n
        $A[k] \leftarrow B[k]$

jeff erickson

5	2	4	7	1	3	2	6

merge-sort (A, p, r)
  if $p < r$
    $q \leftarrow \lfloor (p+r)/2 \rfloor$
    merge-sort (A, p, q)
    merge-sort $(A, q+1, r)$
    merge(A, p, q, r)

MERGE$(A[1..n], m)$:
  $i \leftarrow 1$; $j \leftarrow m+1$
  for $k \leftarrow 1$ to n
    if $j > n$
      $B[k] \leftarrow A[i]$; $i \leftarrow i+1$
    else if $i > m$
      $B[k] \leftarrow A[j]$; $j \leftarrow j+1$
    else if $A[i] < A[j]$
      $B[k] \leftarrow A[i]$; $i \leftarrow i+1$
    else
      $B[k] \leftarrow A[j]$; $j \leftarrow j+1$
  for $k \leftarrow 1$ to n
    $A[k] \leftarrow B[k]$

jeff erickson

5	2	4	7	1	3	2	6

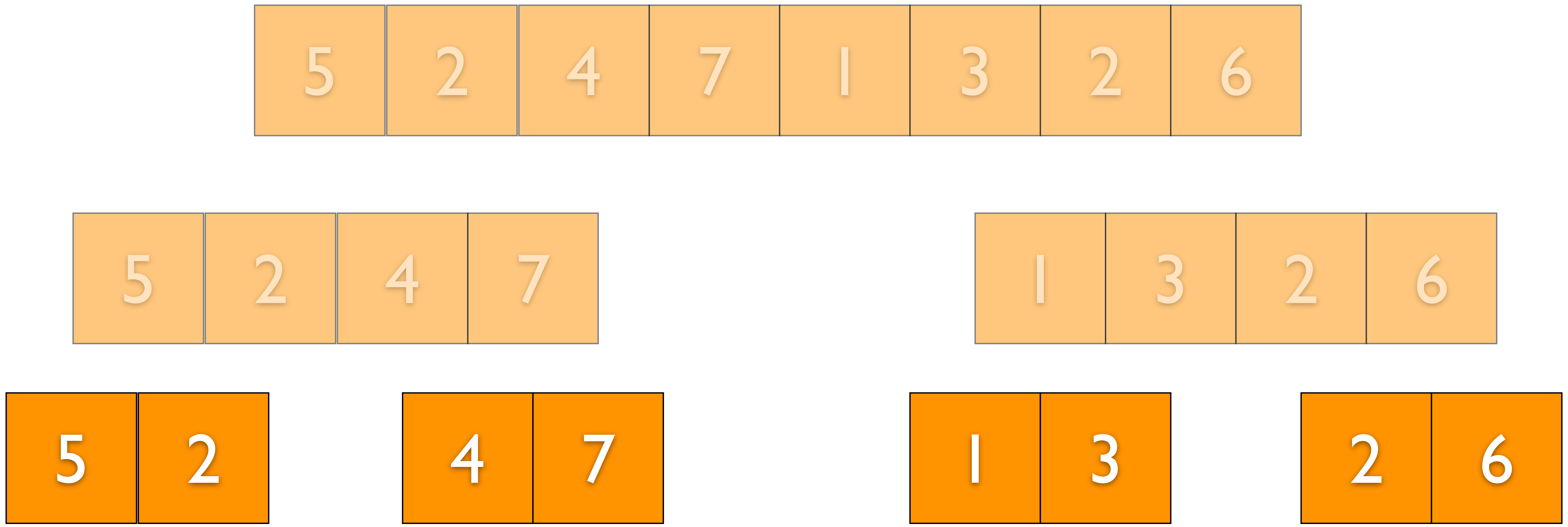
5	2	4	7

1	3	2	6

merge-sort (A, p, r)
  if $p < r$
    $q \leftarrow \lfloor (p+r)/2 \rfloor$
    merge-sort (A, p, q)
    merge-sort $(A, q+1, r)$
    merge(A, p, q, r)

```
MERGE(A[1..n], m):
  i ← 1; j ← m + 1
  for k ← 1 to n
    if j > n
      B[k] ← A[i]; i ← i + 1
    else if i > m
      B[k] ← A[j]; j ← j + 1
    else if A[i] < A[j]
      B[k] ← A[i]; i ← i + 1
    else
      B[k] ← A[j]; j ← j + 1
  for k ← 1 to n
    A[k] ← B[k]
```

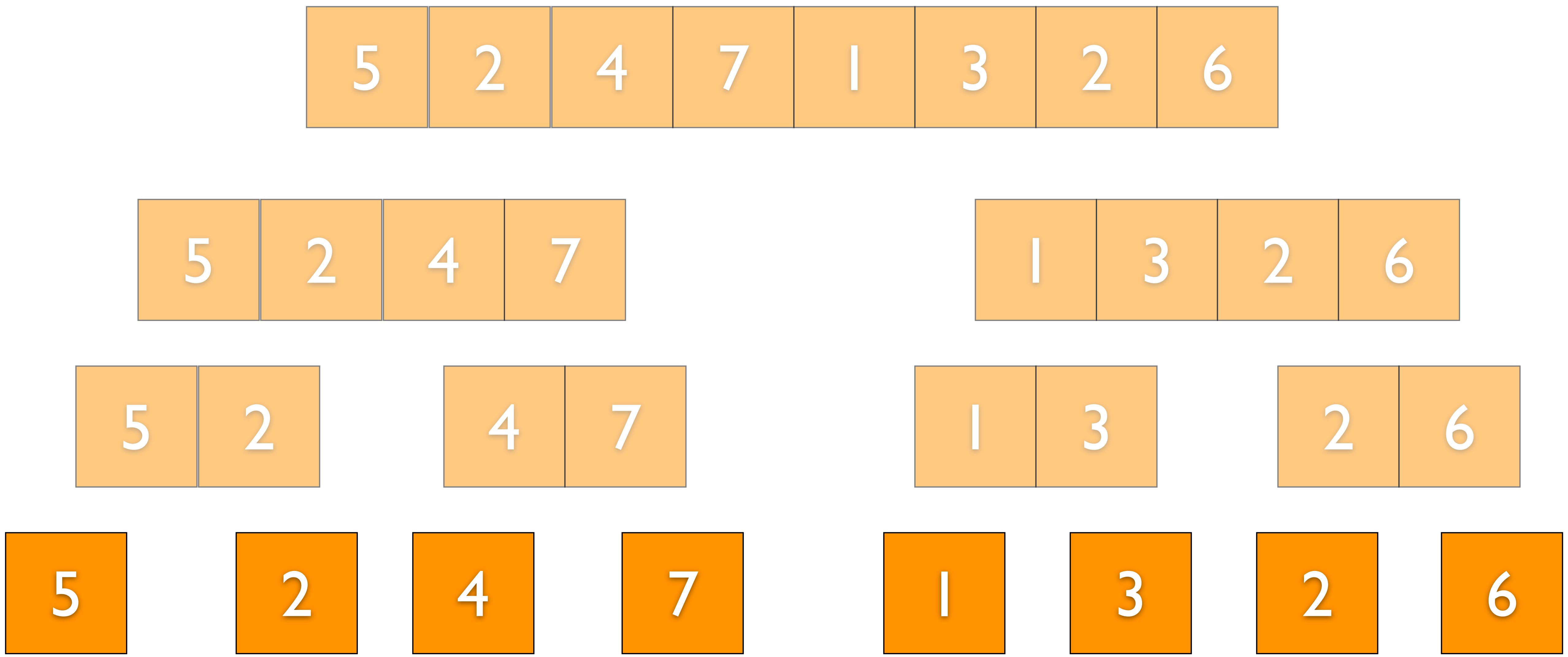
jeff erickson



merge-sort (A, p, r)
  if $p < r$
    $q \leftarrow \lfloor (p+r)/2 \rfloor$
    merge-sort (A, p, q)
    merge-sort $(A, q+1, r)$
    merge(A, p, q, r)

MERGE$(A[1..n], m)$:
  $i \leftarrow 1$; $j \leftarrow m+1$
  for $k \leftarrow 1$ to n
    if $j > n$
      $B[k] \leftarrow A[i]$; $i \leftarrow i+1$
    else if $i > m$
      $B[k] \leftarrow A[j]$; $j \leftarrow j+1$
    else if $A[i] < A[j]$
      $B[k] \leftarrow A[i]$; $i \leftarrow i+1$
    else
      $B[k] \leftarrow A[j]$; $j \leftarrow j+1$
  for $k \leftarrow 1$ to n
    $A[k] \leftarrow B[k]$

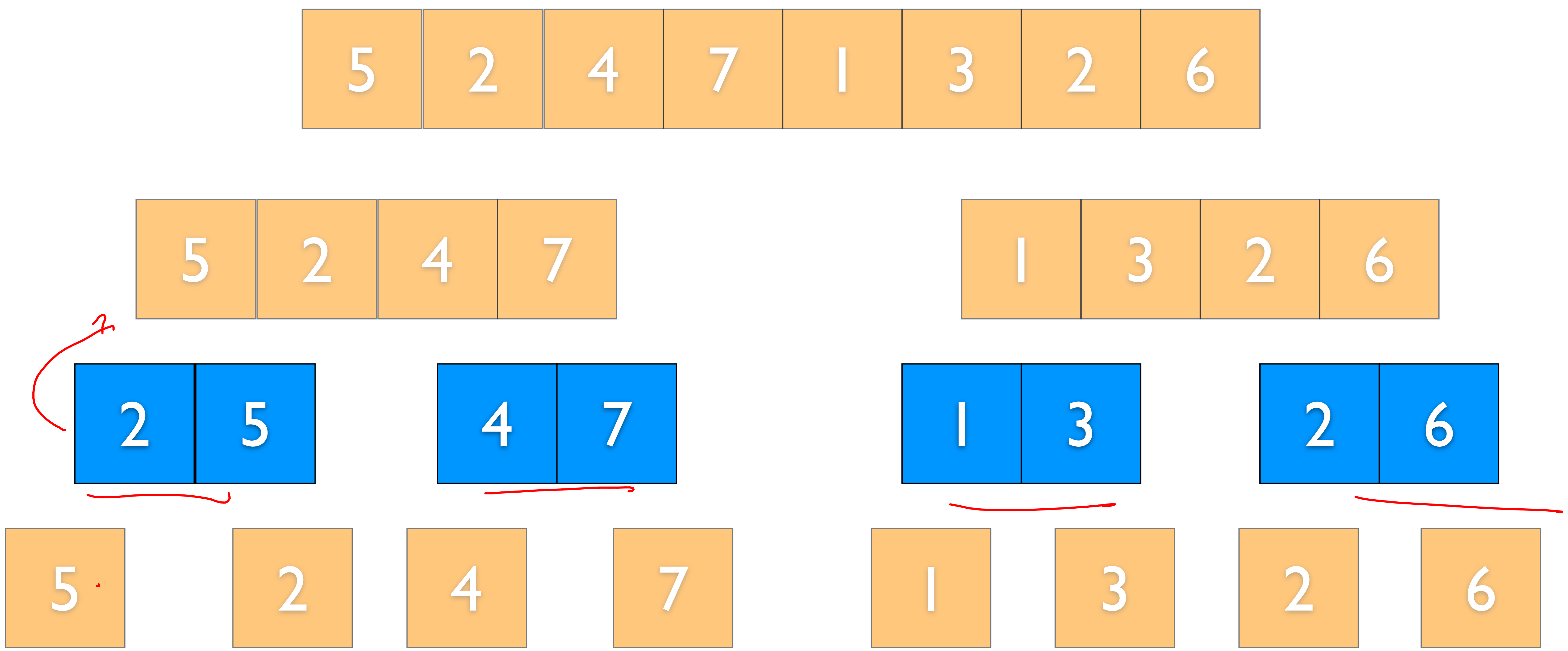
jeff erickson



merge-sort (A, p, r)
  if $p < r$
    $q \leftarrow \lfloor (p + r)/2 \rfloor$
    merge-sort (A, p, q)
    merge-sort $(A, q + 1, r)$
    merge(A, p, q, r)

<u>MERGE$(A[1..n], m)$:</u>
  $i \leftarrow 1;\ j \leftarrow m + 1$
  for $k \leftarrow 1$ to n
    if $j > n$
      $B[k] \leftarrow A[i];\ i \leftarrow i + 1$
    else if $i > m$
      $B[k] \leftarrow A[j];\ j \leftarrow j + 1$
    else if $A[i] < A[j]$
      $B[k] \leftarrow A[i];\ i \leftarrow i + 1$
    else
      $B[k] \leftarrow A[j];\ j \leftarrow j + 1$
  for $k \leftarrow 1$ to n
    $A[k] \leftarrow B[k]$

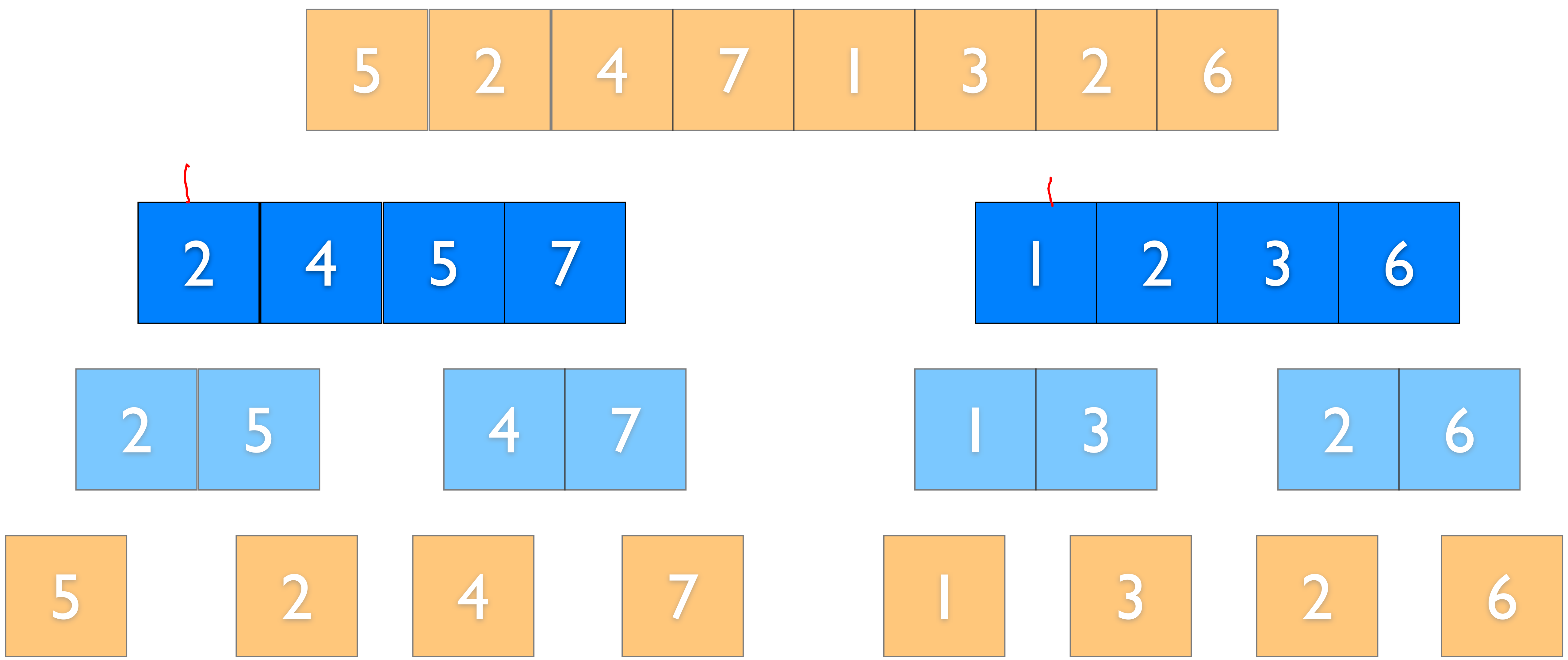
jeff erickson



merge-sort (A, p, r)
  if $p < r$
    $q \leftarrow \lfloor (p+r)/2 \rfloor$
    merge-sort (A, p, q)
    merge-sort $(A, q+1, r)$
    merge(A, p, q, r)

MERGE$(A[1..n], m)$:
  $i \leftarrow 1$; $j \leftarrow m+1$
  for $k \leftarrow 1$ to n
    if $j > n$
      $B[k] \leftarrow A[i]$; $i \leftarrow i+1$
    else if $i > m$
      $B[k] \leftarrow A[j]$; $j \leftarrow j+1$
    else if $A[i] < A[j]$
      $B[k] \leftarrow A[i]$; $i \leftarrow i+1$
    else
      $B[k] \leftarrow A[j]$; $j \leftarrow j+1$
  for $k \leftarrow 1$ to n
    $A[k] \leftarrow B[k]$

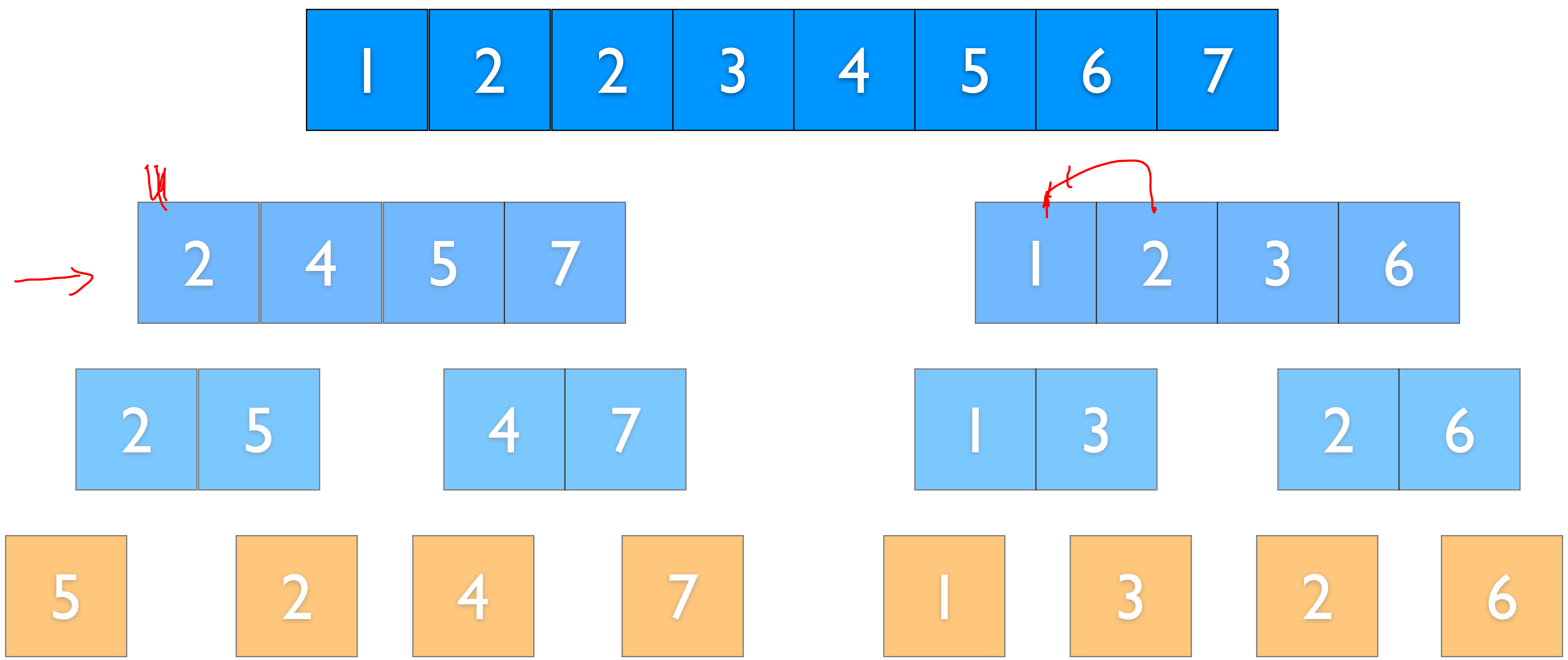
jeff erickson



merge-sort (A, p, r)
  if $p < r$
    $q \leftarrow \lfloor (p + r)/2 \rfloor$
    merge-sort (A, p, q)
    merge-sort $(A, q + 1, r)$
    merge(A, p, q, r)

MERGE$(A[1..n], m)$:
  $i \leftarrow 1$; $j \leftarrow m + 1$
  for $k \leftarrow 1$ to n
    if $j > n$
      $B[k] \leftarrow A[i]$; $i \leftarrow i + 1$
    else if $i > m$
      $B[k] \leftarrow A[j]$; $j \leftarrow j + 1$
    else if $A[i] < A[j]$
      $B[k] \leftarrow A[i]$; $i \leftarrow i + 1$
    else
      $B[k] \leftarrow A[j]$; $j \leftarrow j + 1$
  for $k \leftarrow 1$ to n
    $A[k] \leftarrow B[k]$

jeff erickson



merge-sort (A, p, r)

  if $p < r$

    $q \leftarrow \lfloor (p + r)/2 \rfloor$ — O(1)

    merge-sort (A, p, q) — $T(n/2)$

    merge-sort $(A, q+1, r)$ — $T(n/2)$

    merge(A, p, q, r) → $\Theta(n)$

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

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

by Masters

merge-sort (A, p, r)

  if $p < r$

    $q \leftarrow \lfloor (p + r)/2 \rfloor$

    merge-sort (A, p, q)

    merge-sort $(A, q + 1, r)$

    merge(A, p, q, r)

closest pair

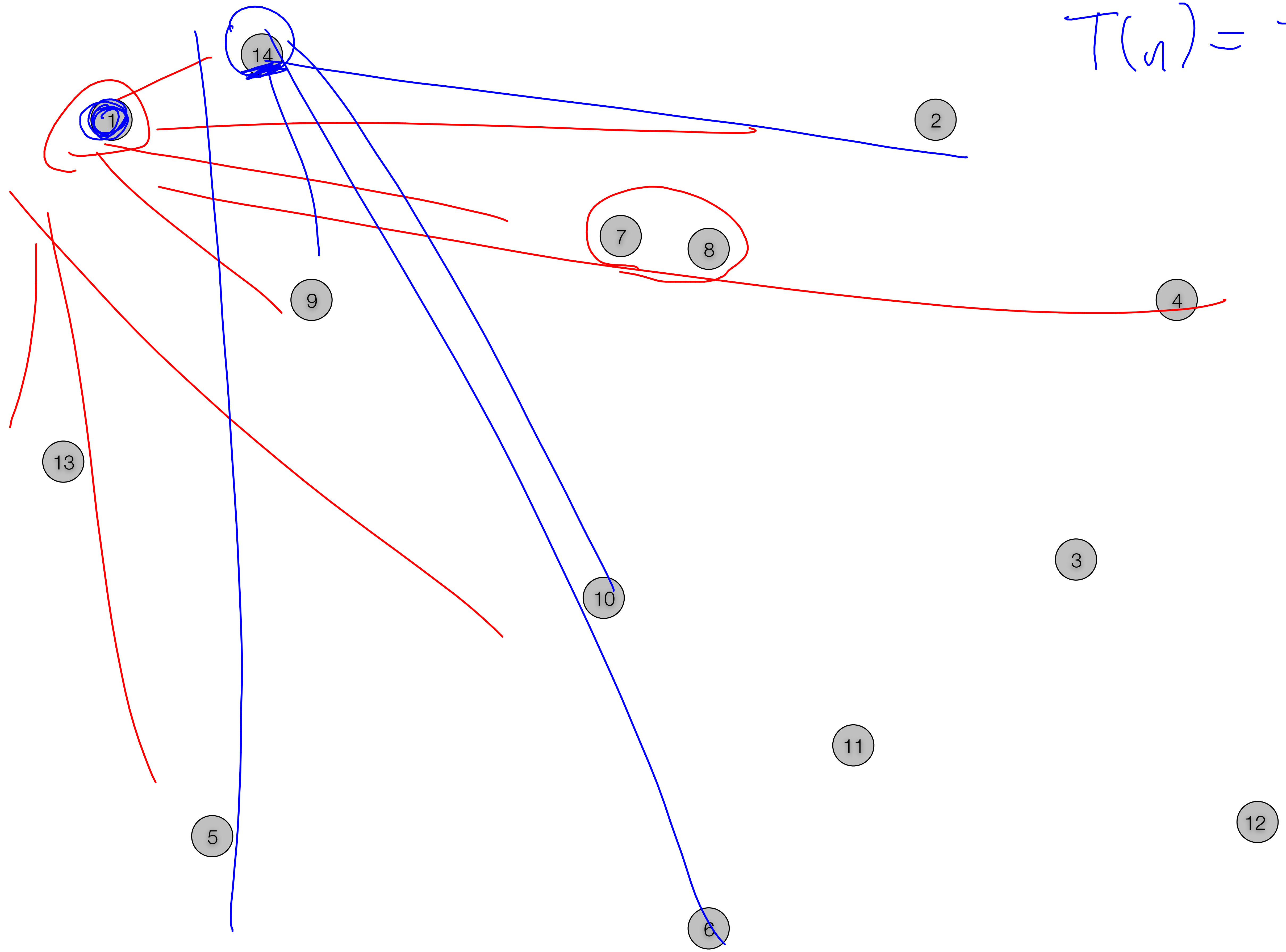
of points

simple brute force approach takes

n pts

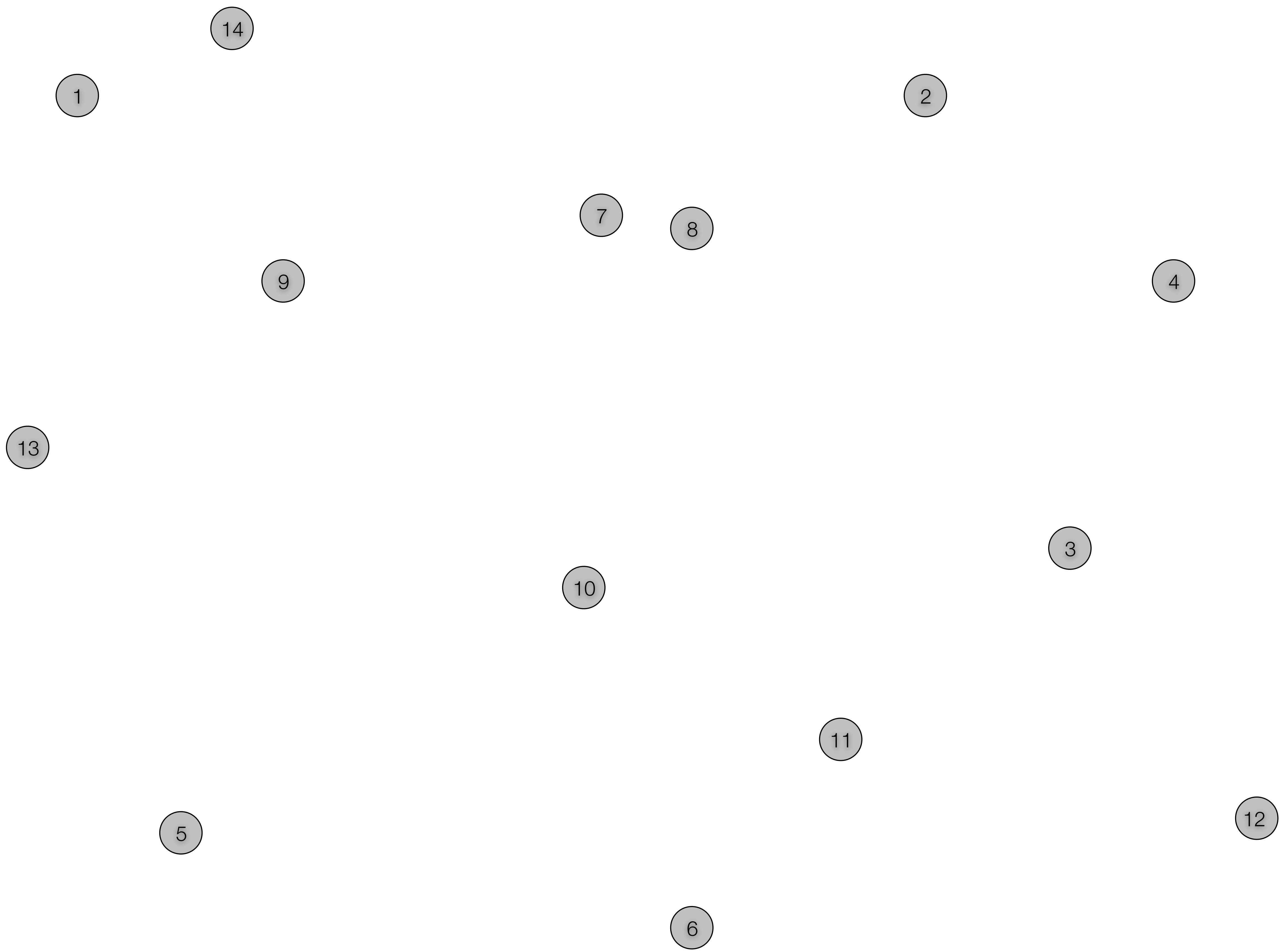
$$T(n) = T(n-1) + n.$$

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

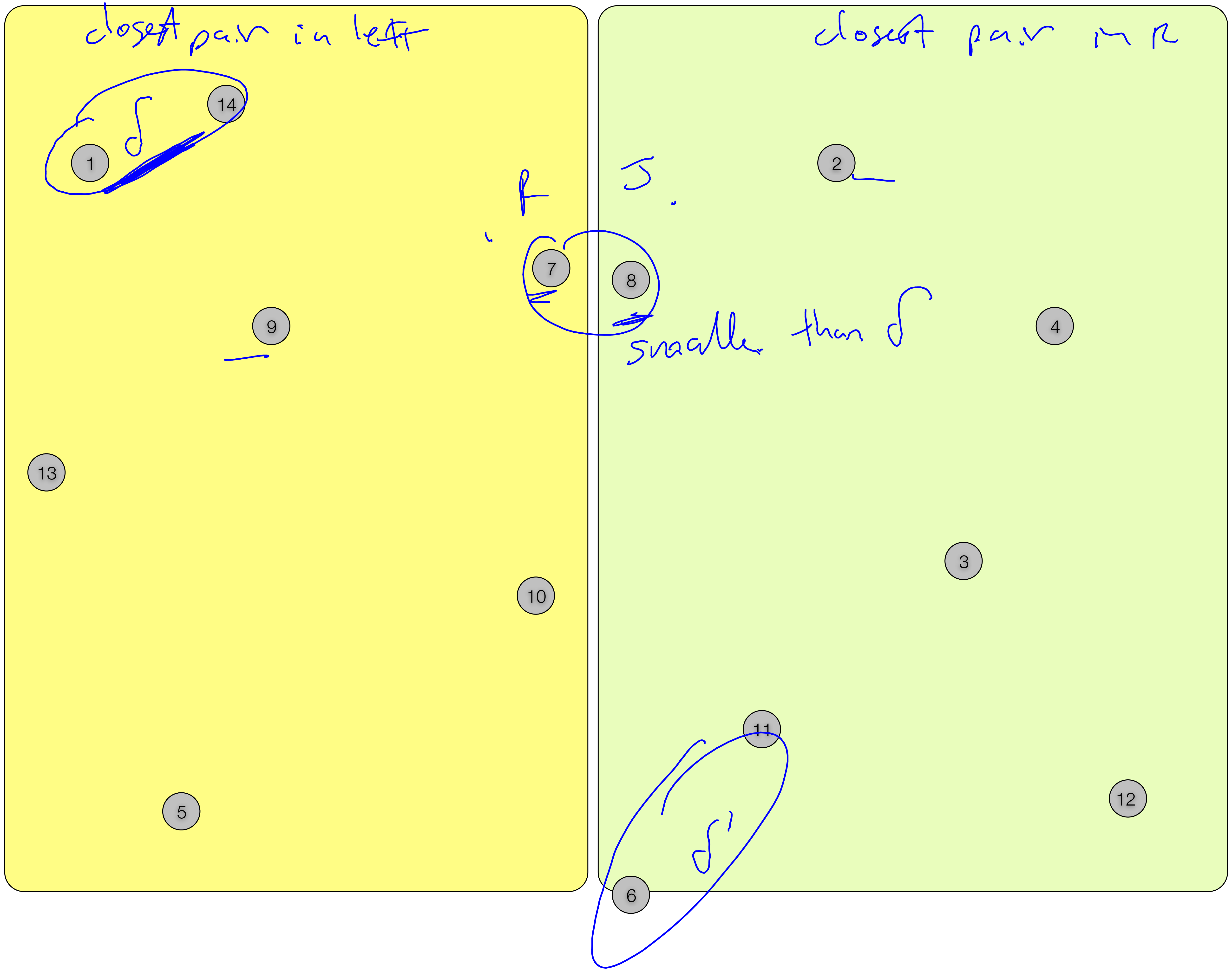


solve the large problem by

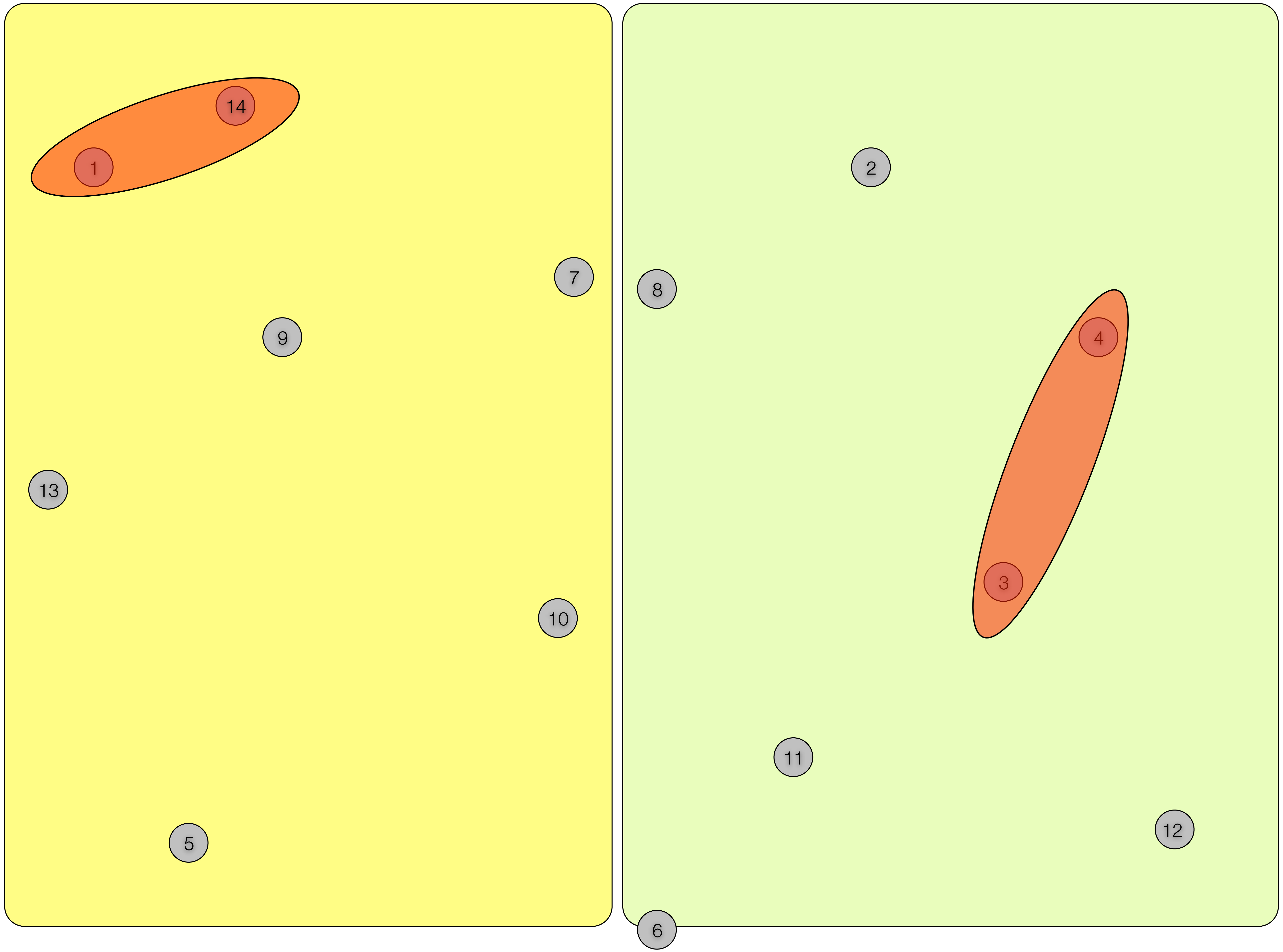
solving smaller problems
and combining solutions


14
1
2
7
8
9
4
13
3
10
11
5
12
6

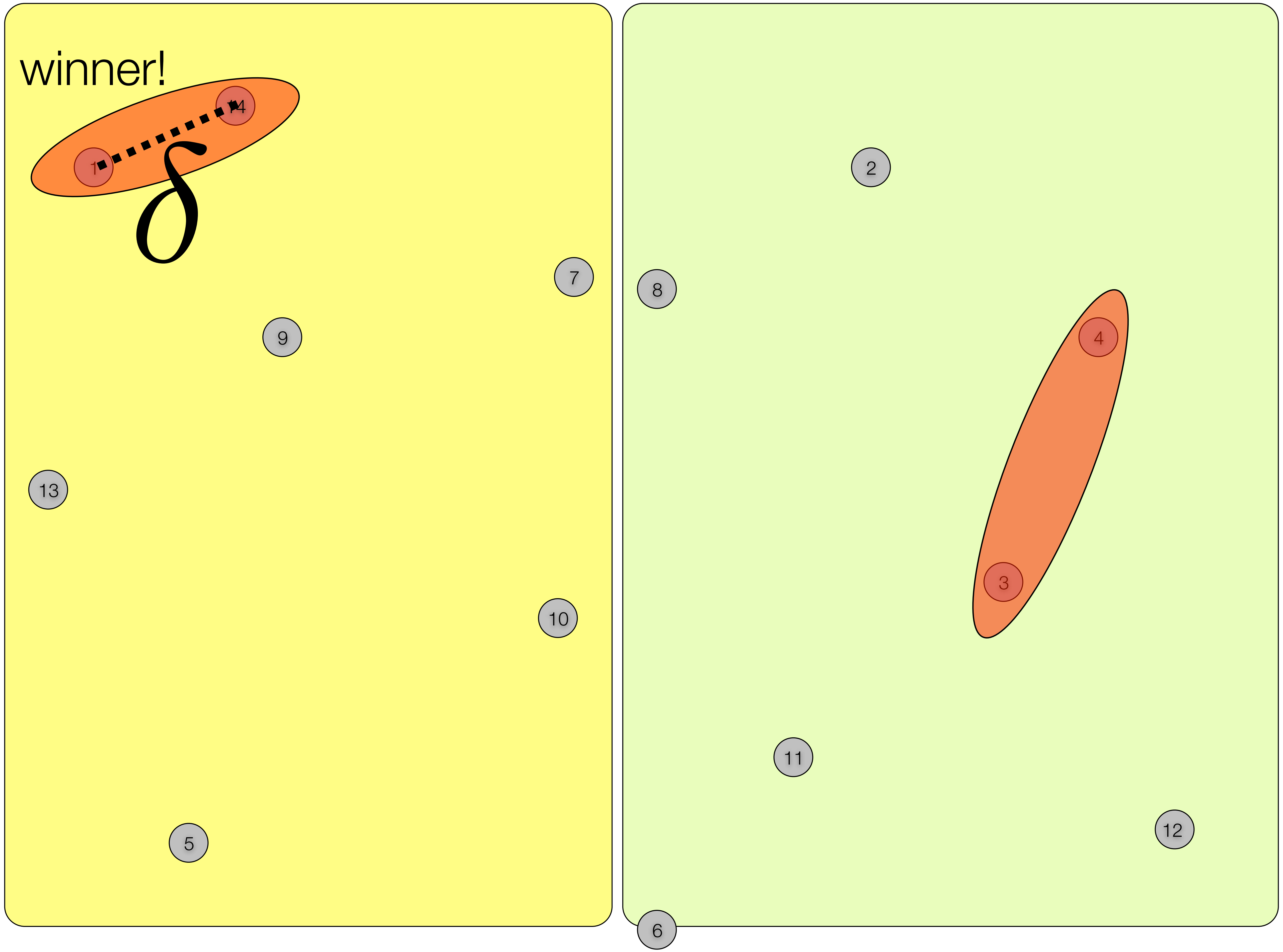
Divide & Conquer

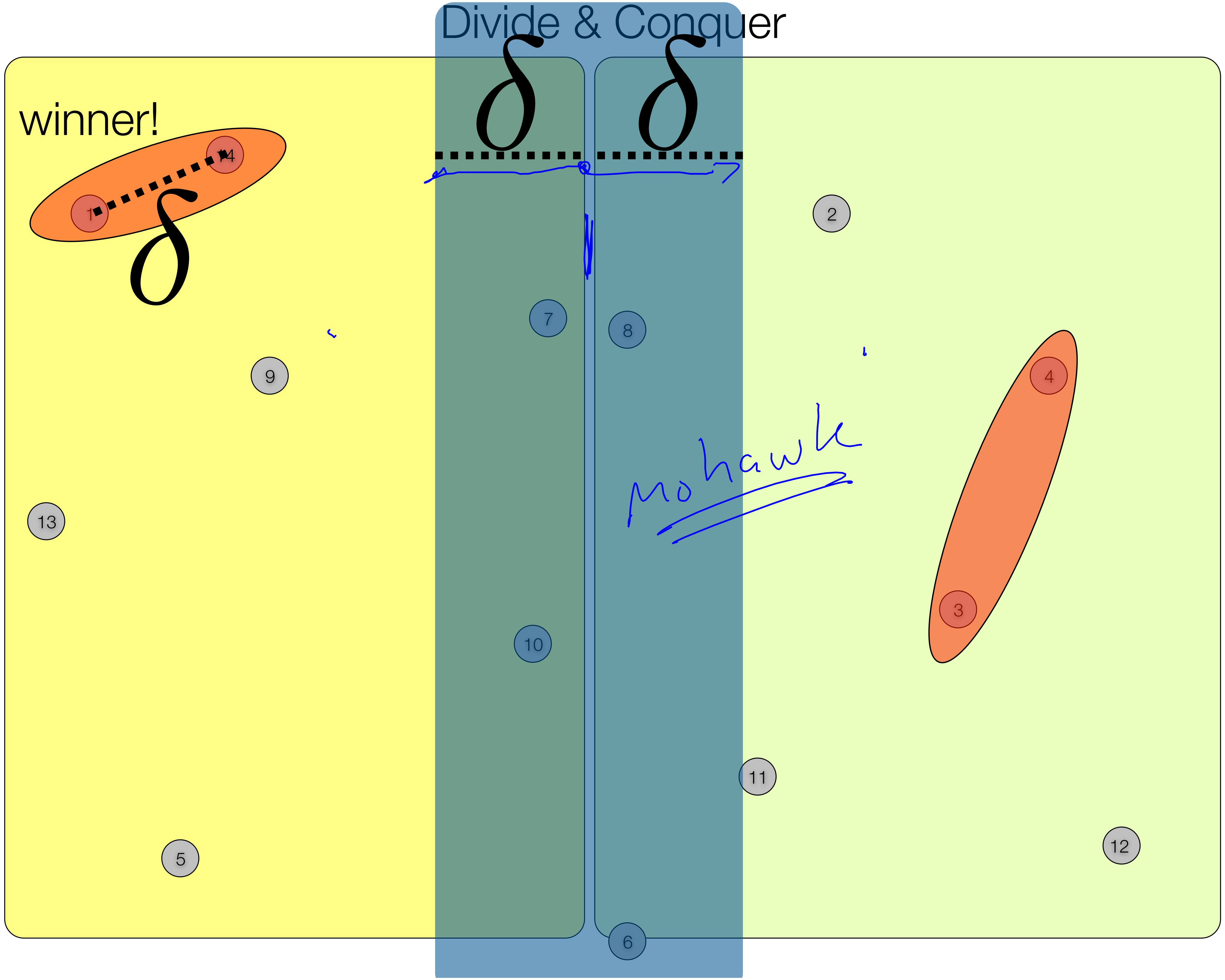


Divide & Conquer

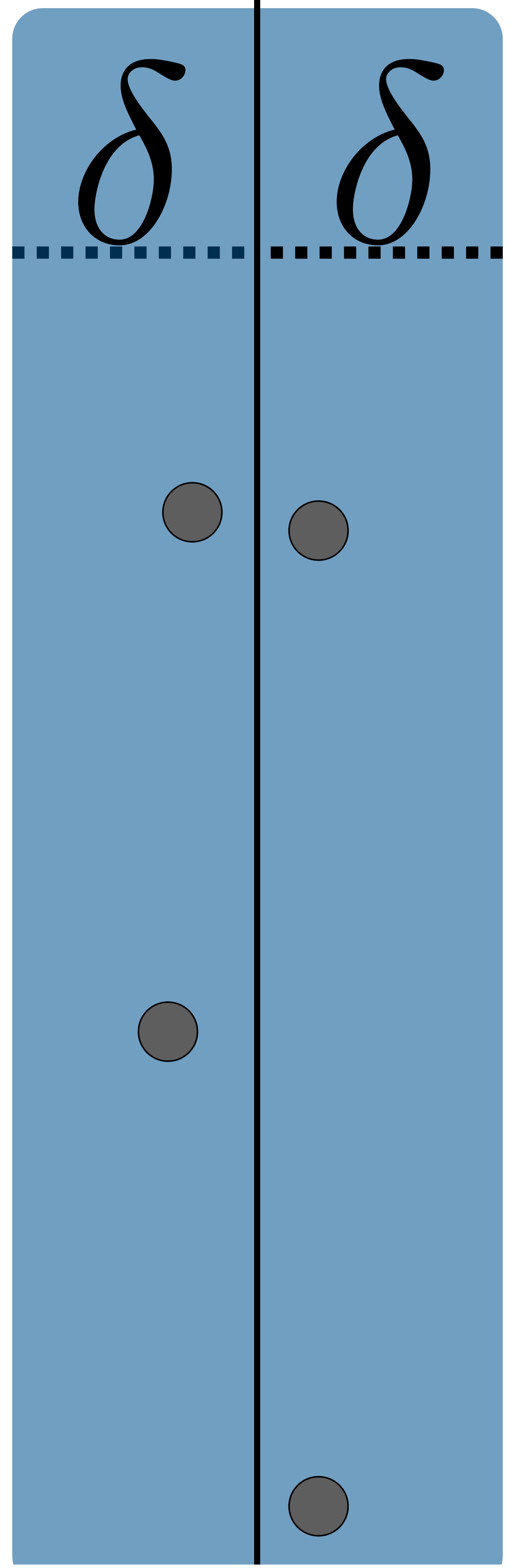


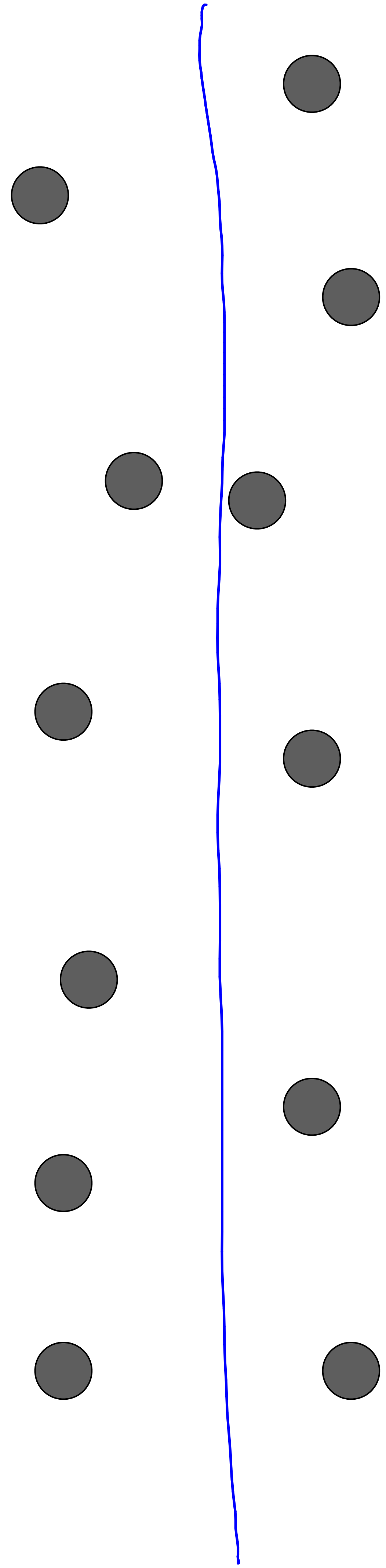
Divide & Conquer





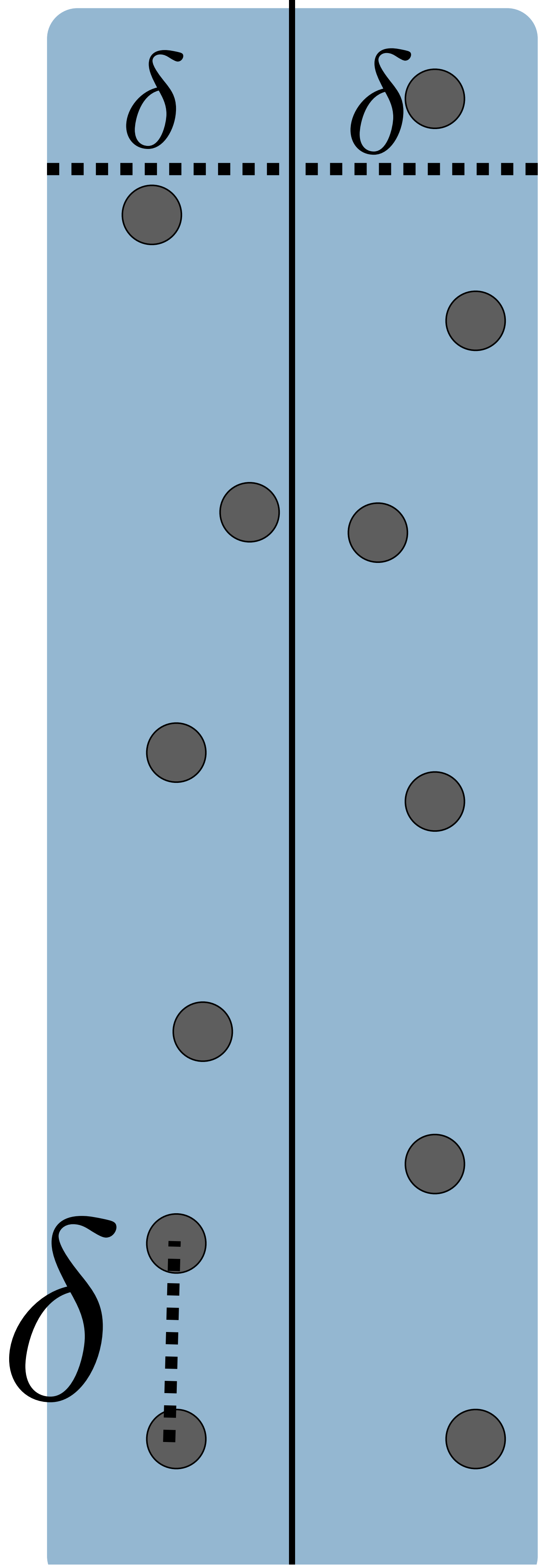
Divide & Conquer
winner!
δ
δ
δ
mohawk
1
14
2
9
7
8
13
4
3
10
11
5
12
6

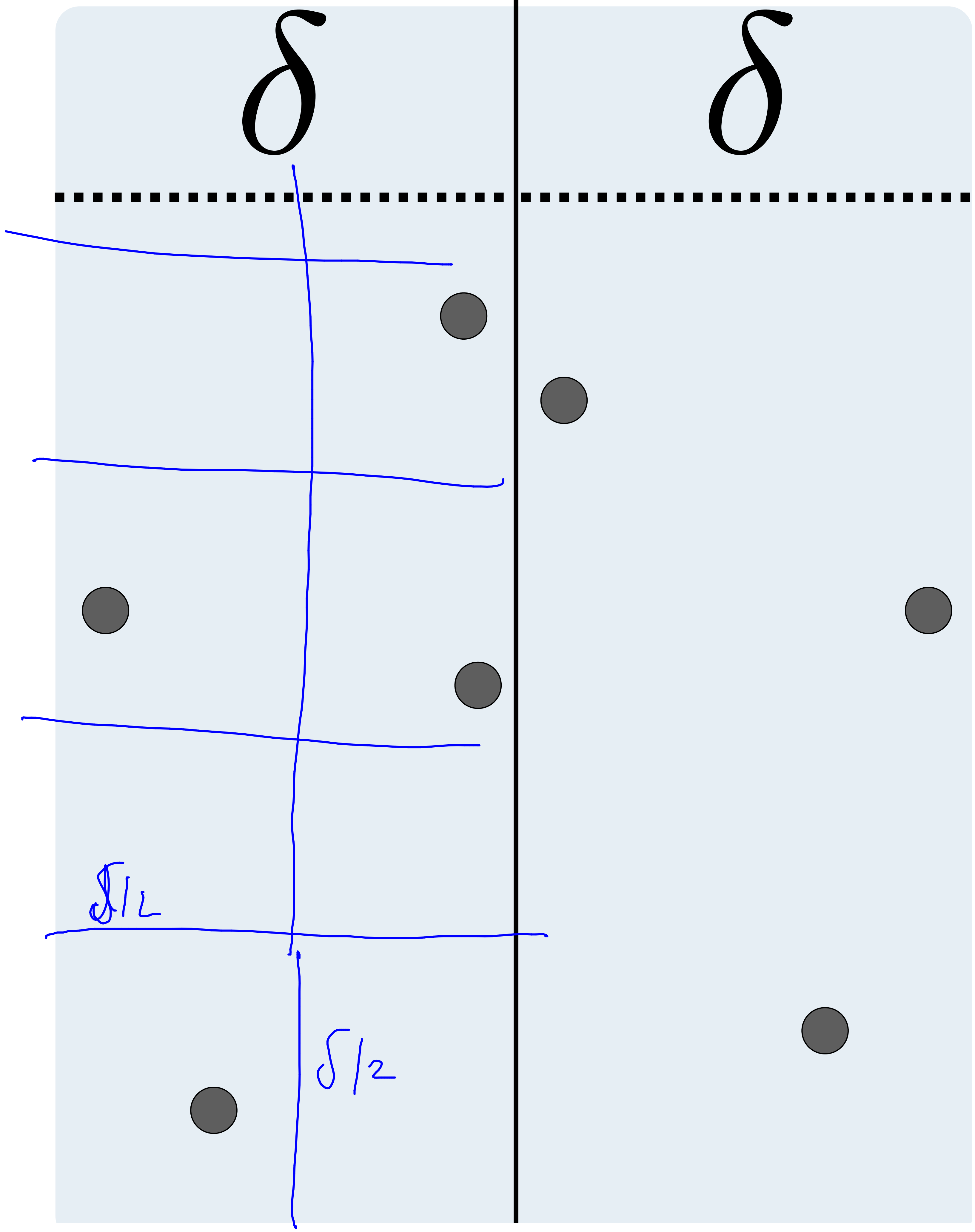

δ
δ



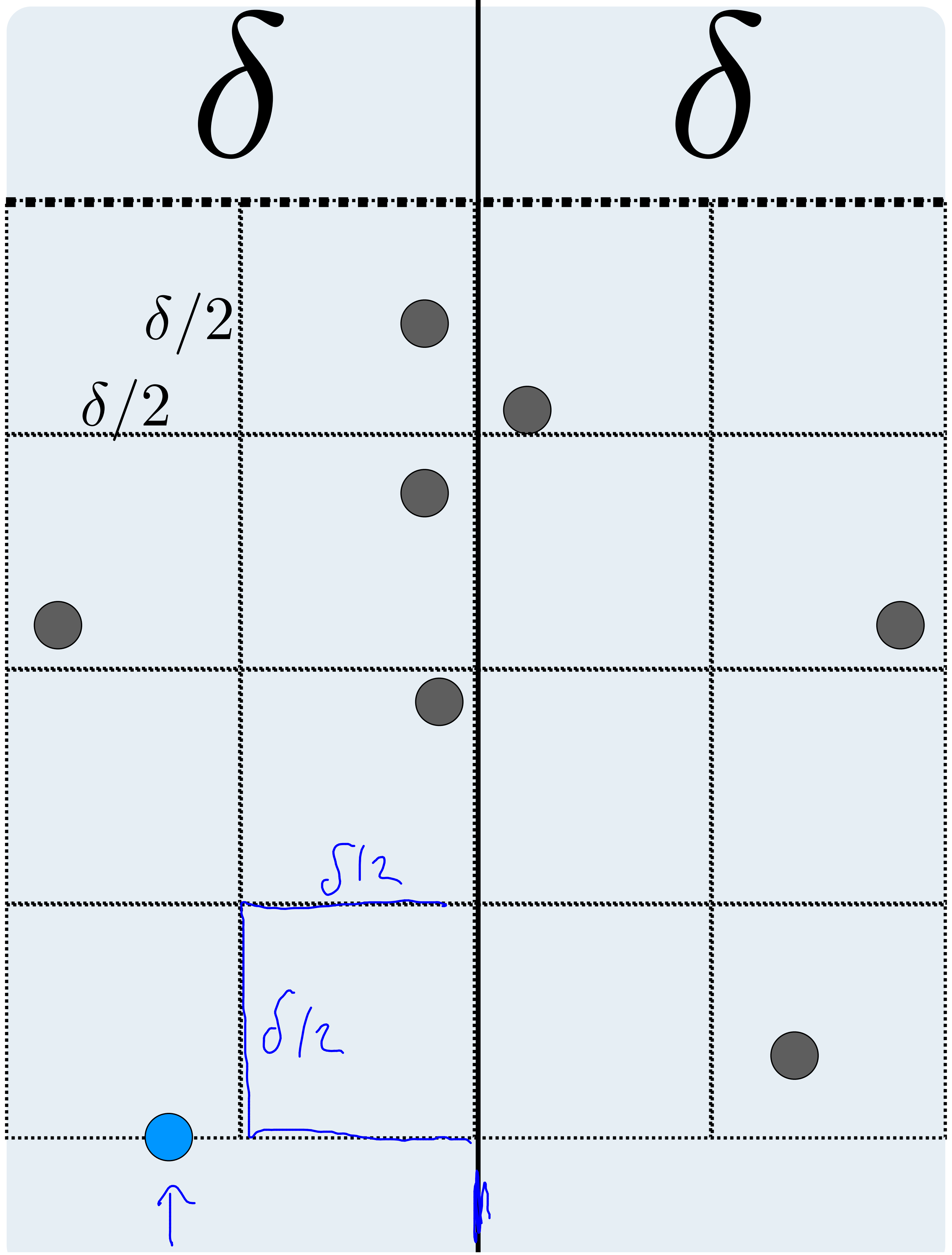
→ all n pts are in the Mohawk!!

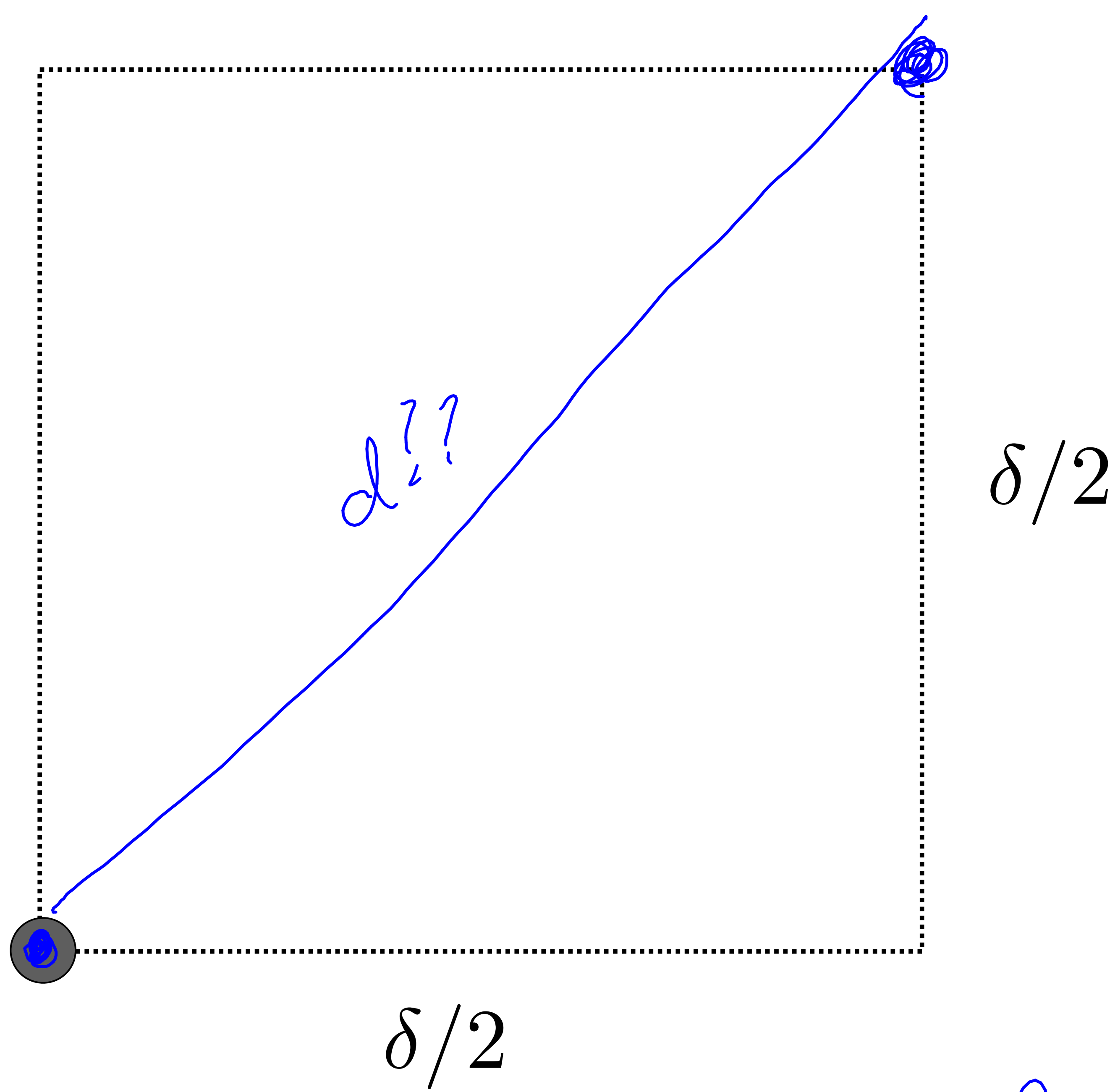
what to do??


δ
δ
δ


δ
δ

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



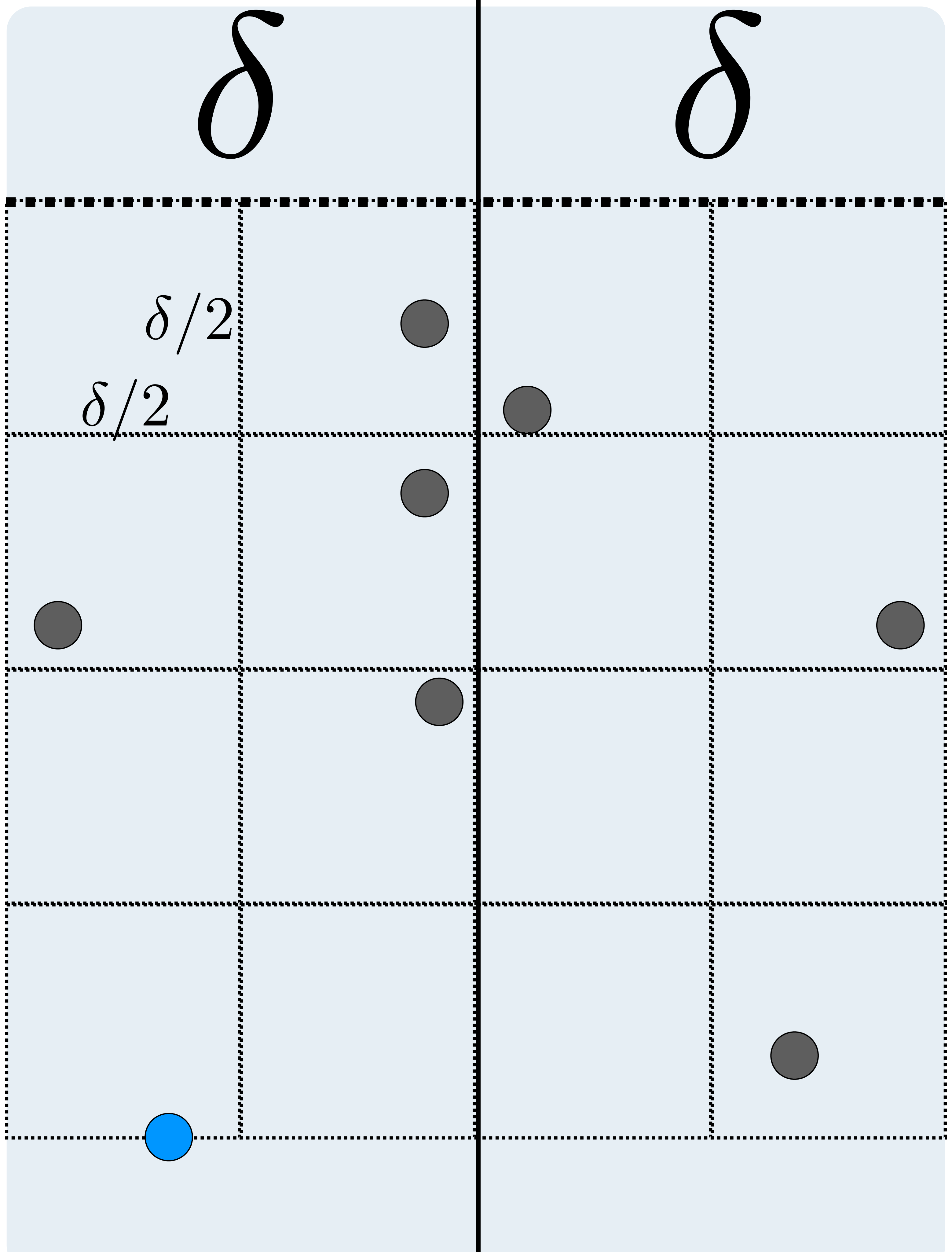


~~at most 1 pt per cubby.~~

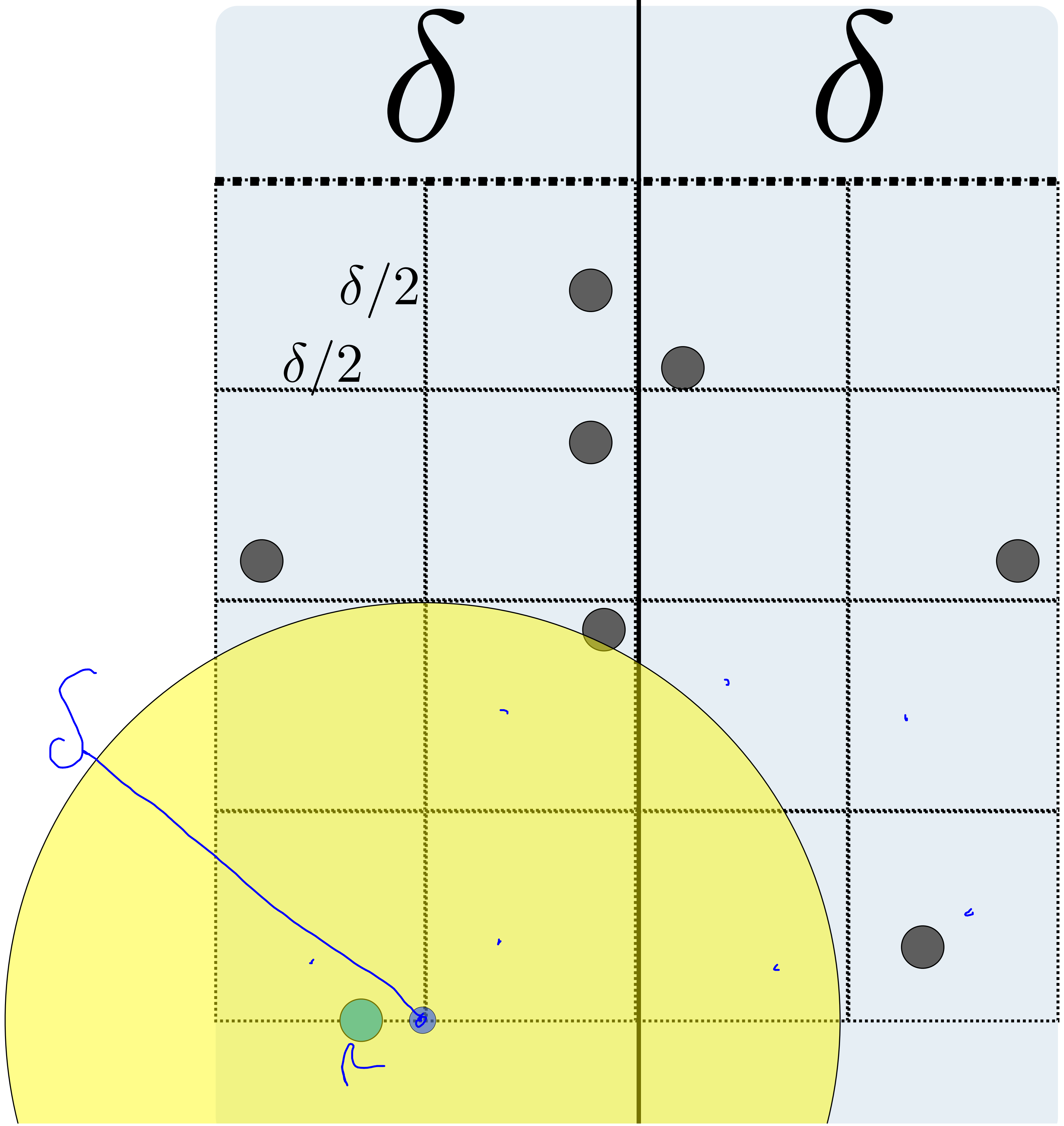
$$\frac{\sqrt{2}}{2} \cdot \delta < \delta$$

this can't happen

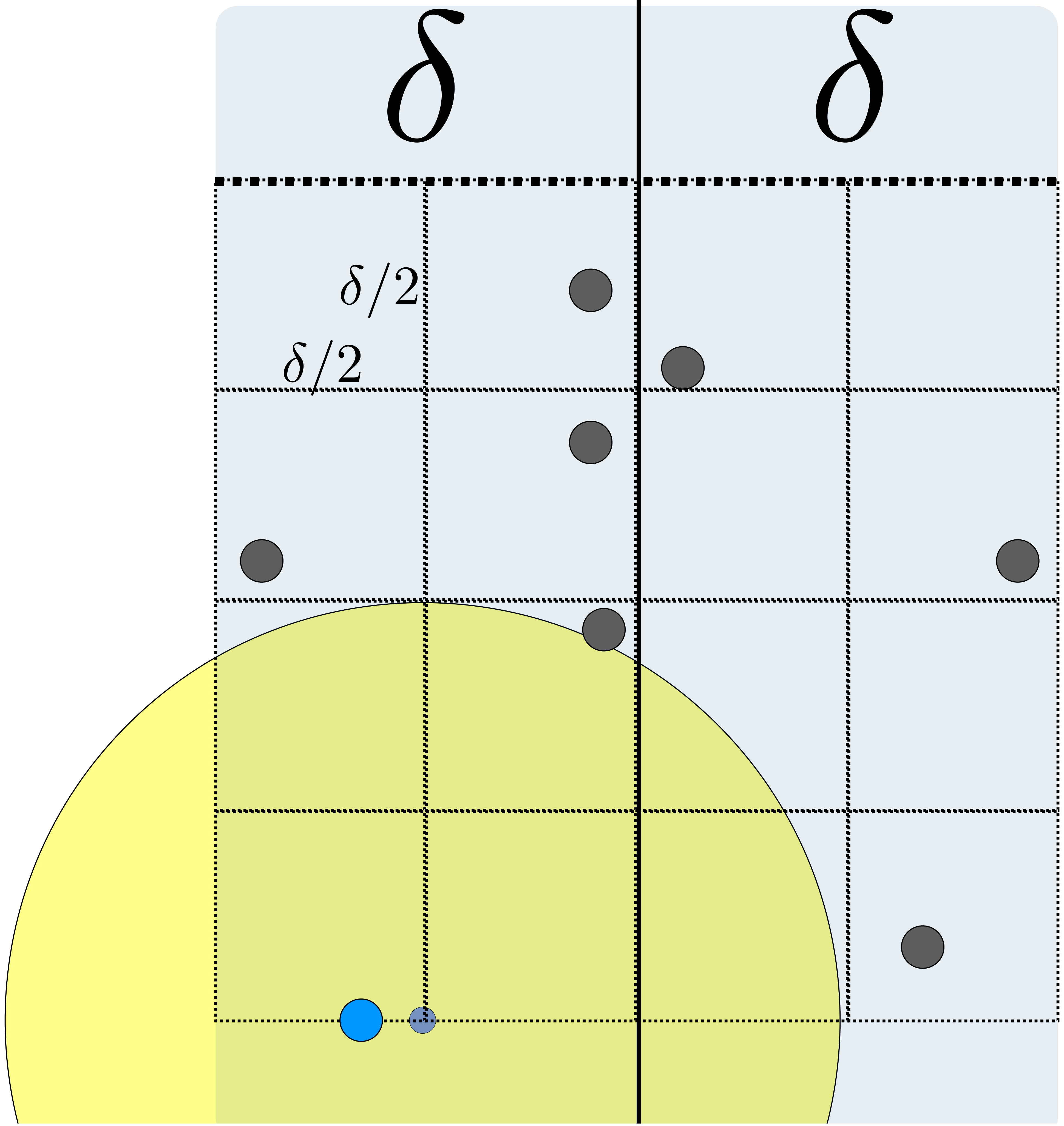
lonely cubby property



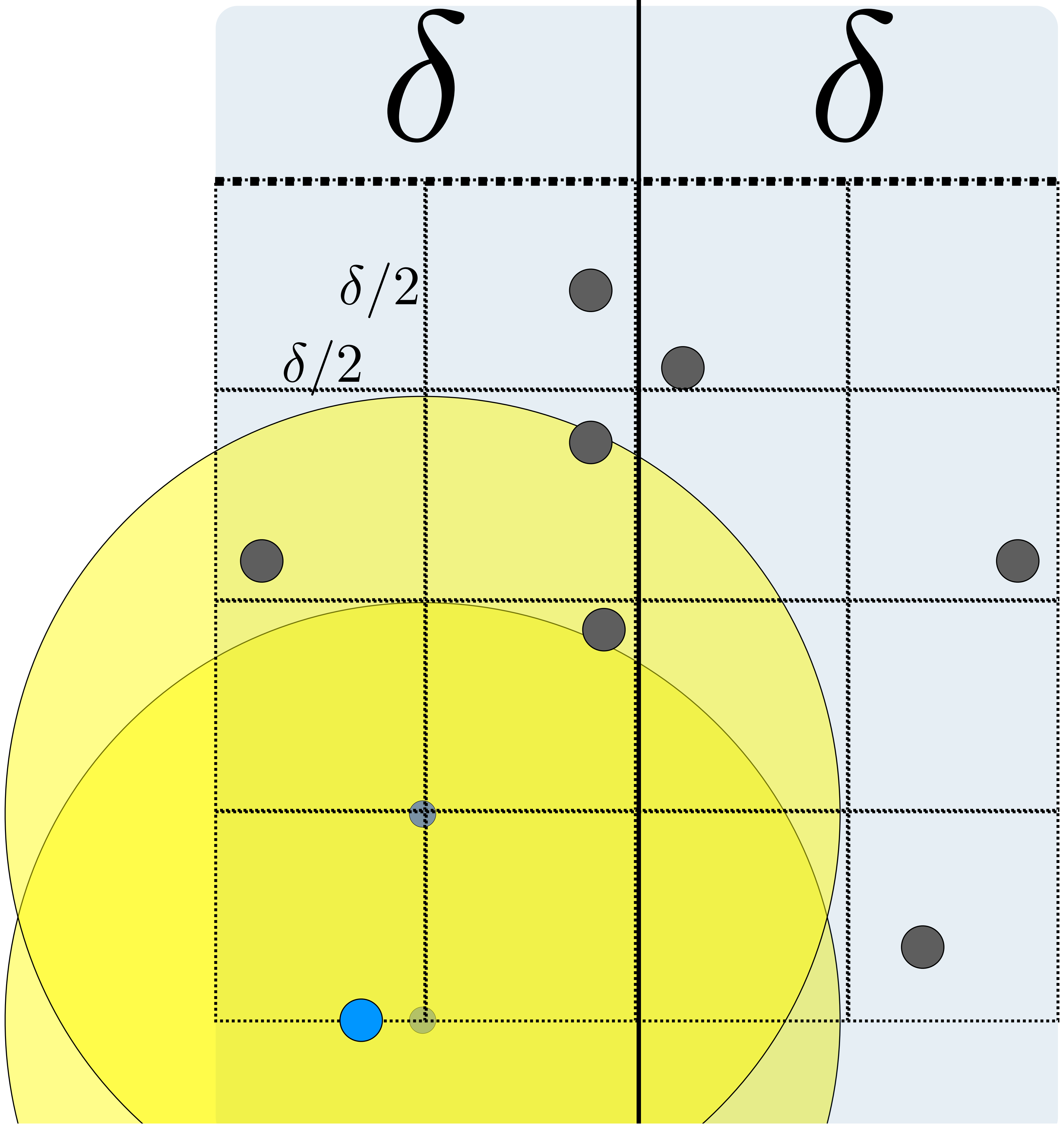
FACT: At most 1 point in each cubby



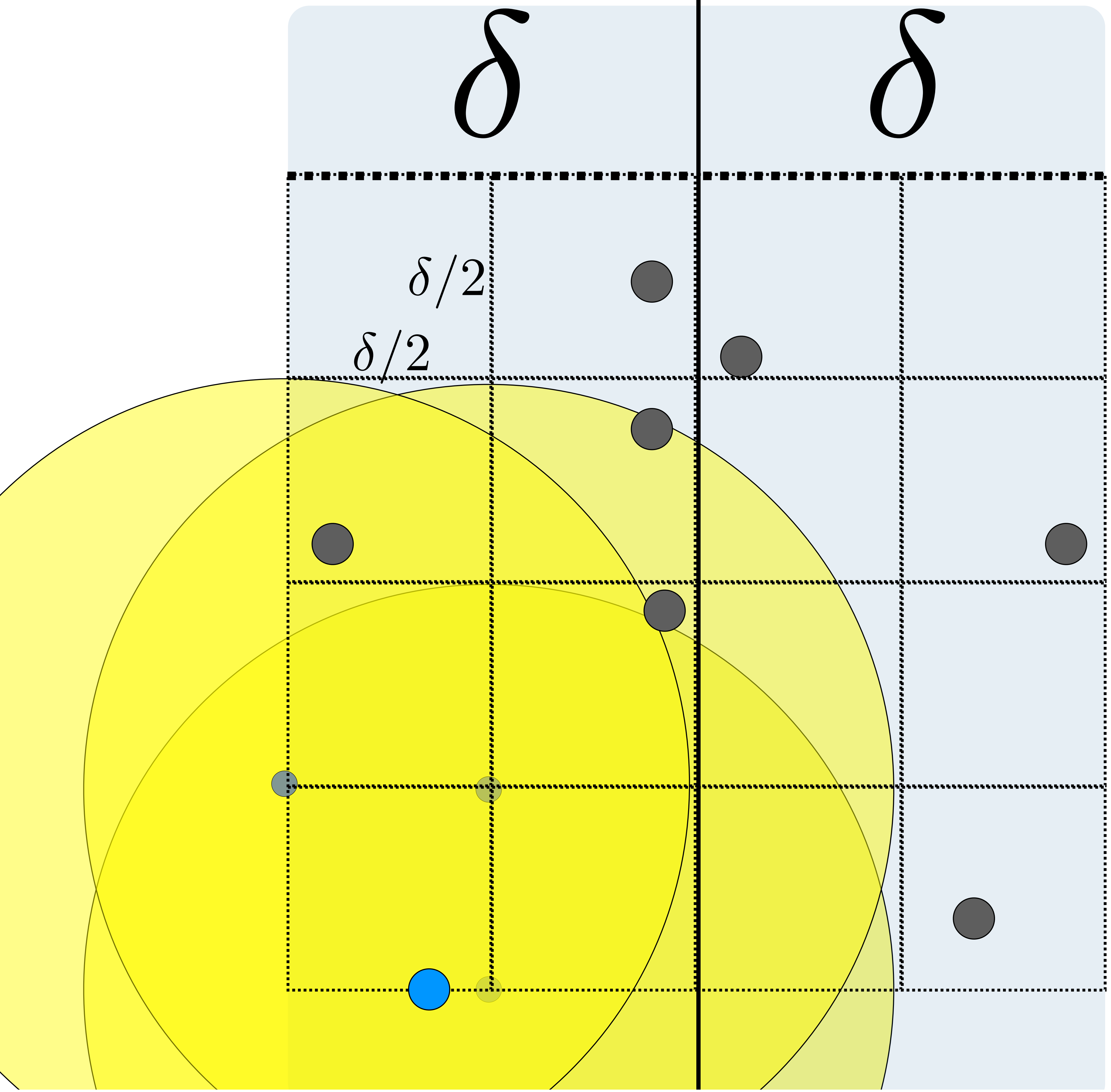
FACT: <=1 point per cubby



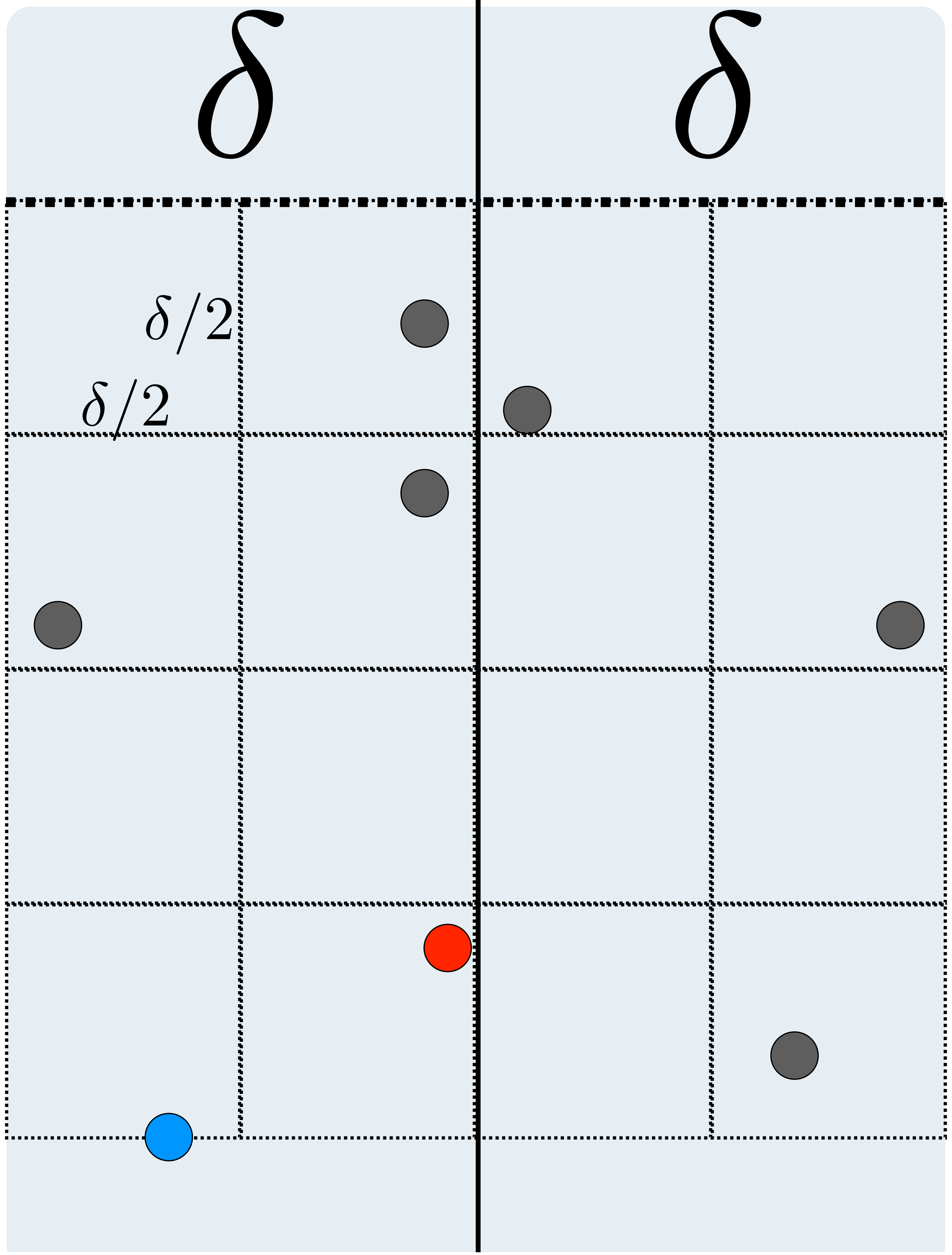
FACT: <=1 point per cubby

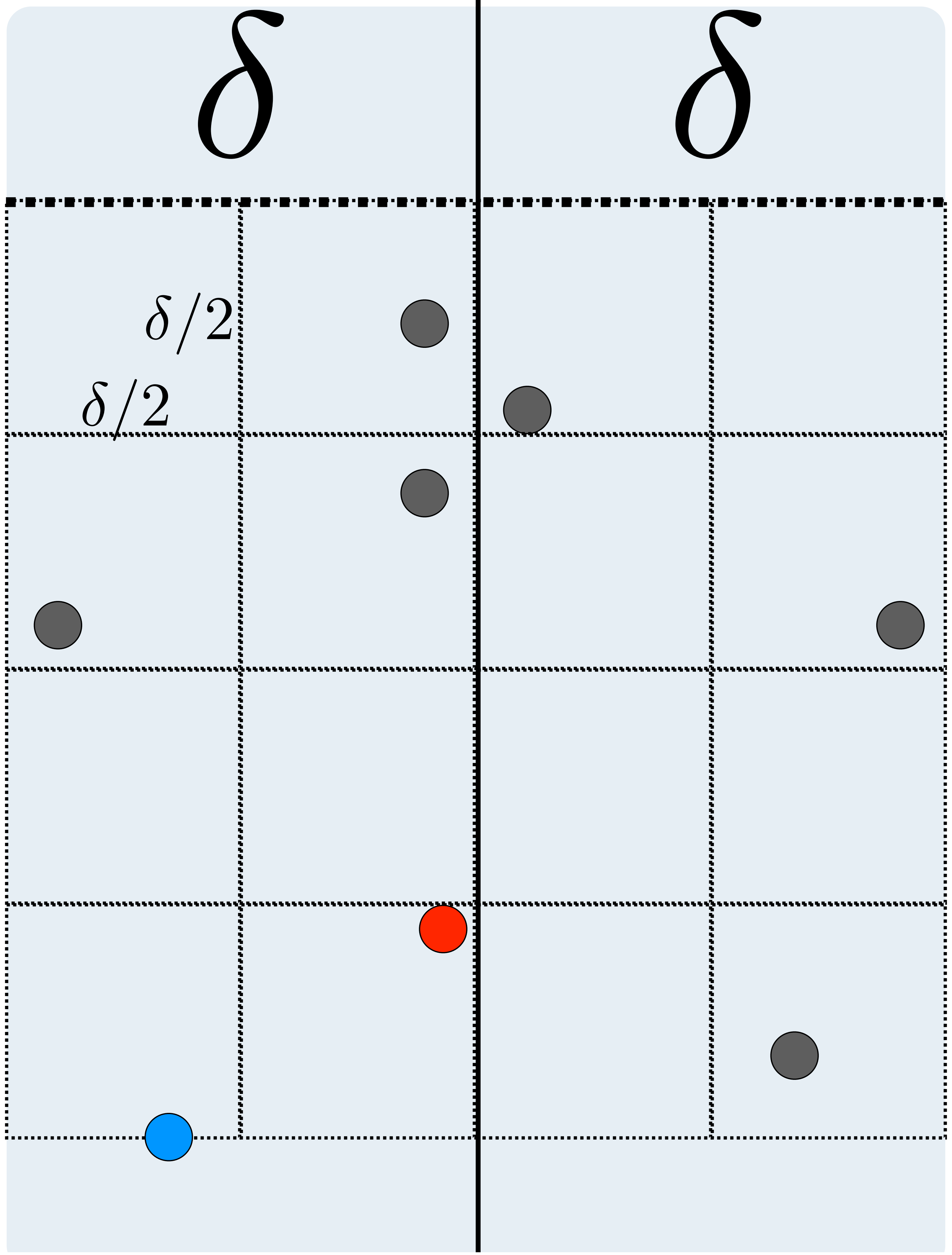


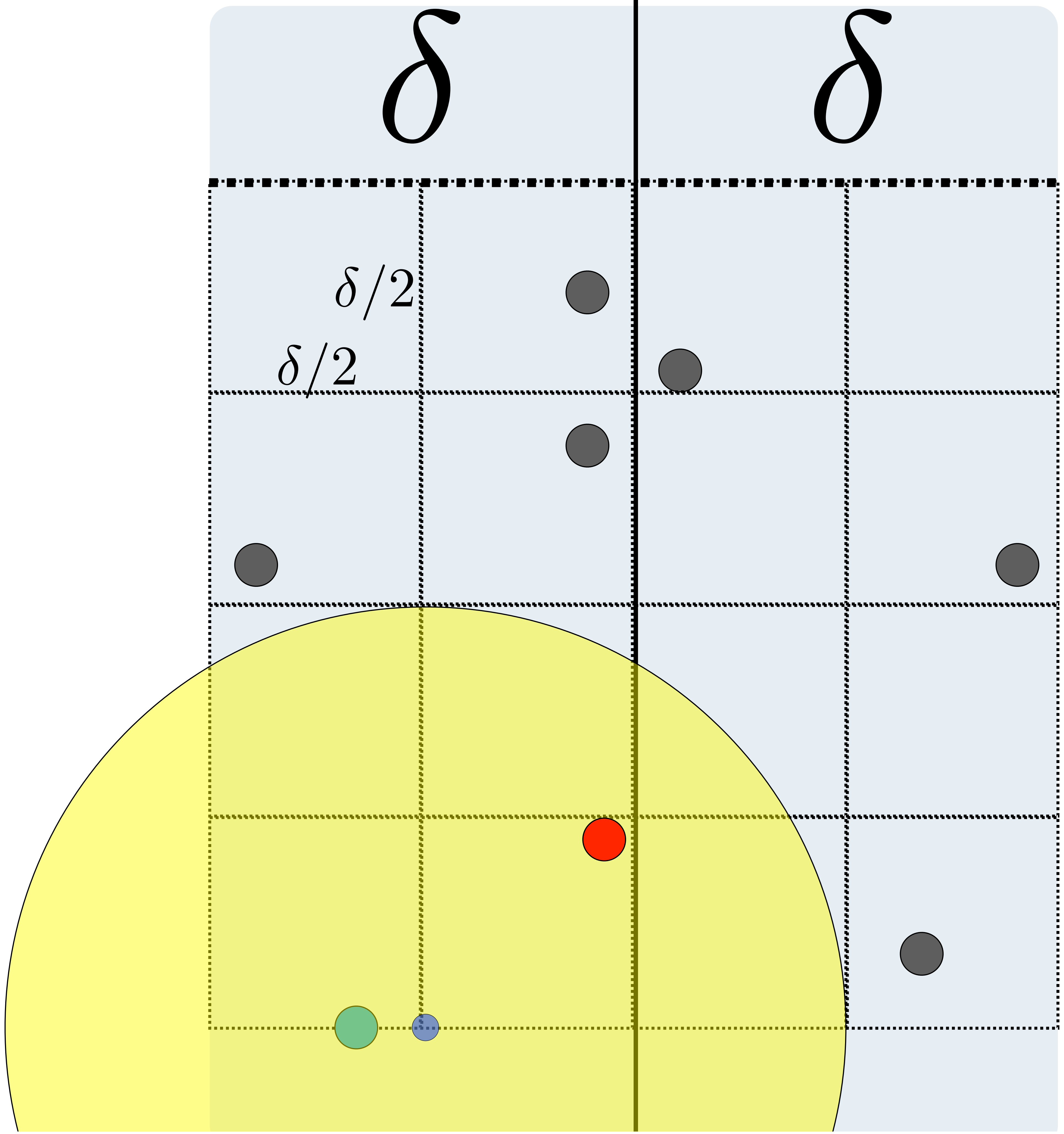
FACT: <=1 point per cubby

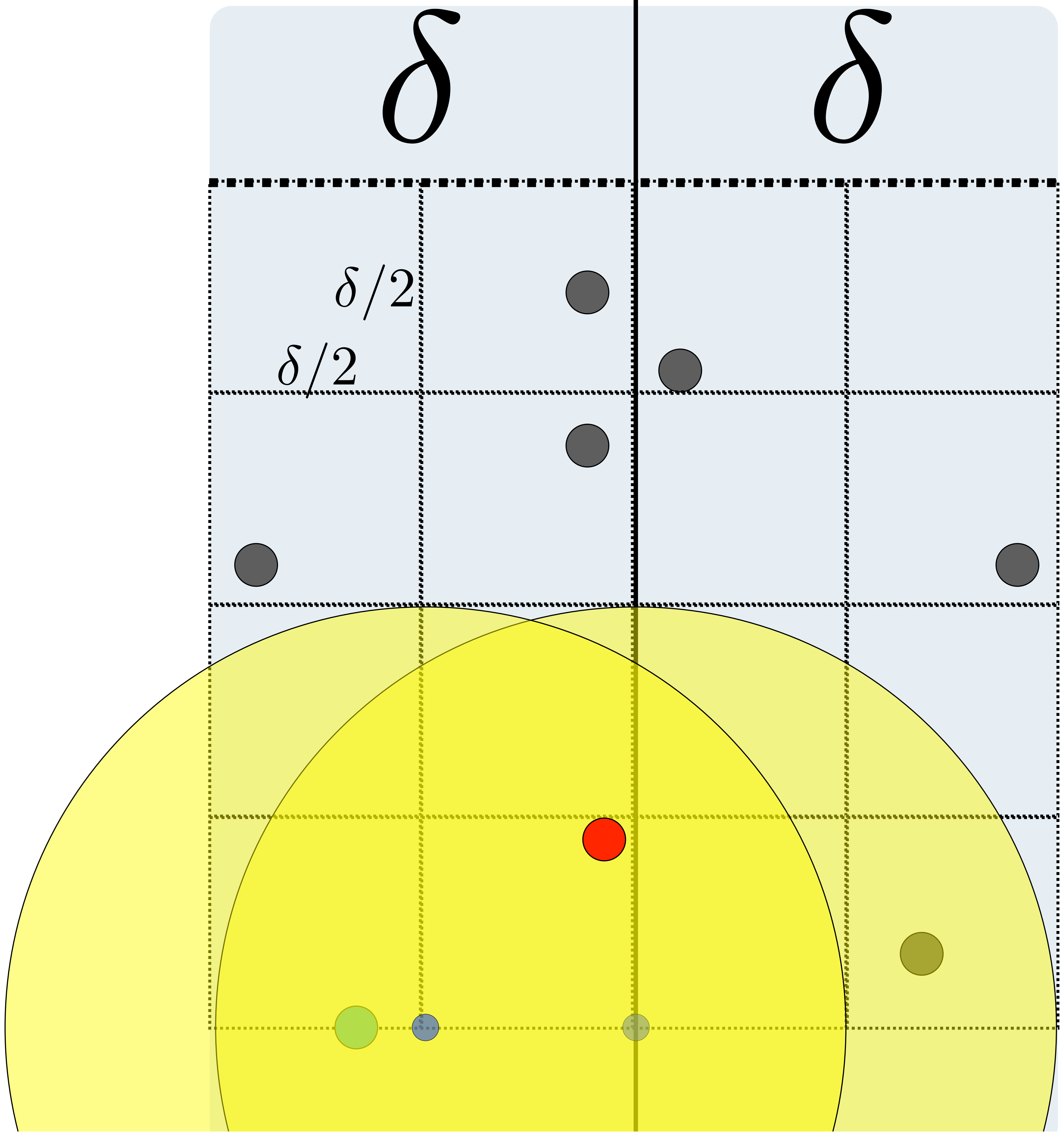


FACT: <=1 point per cubby

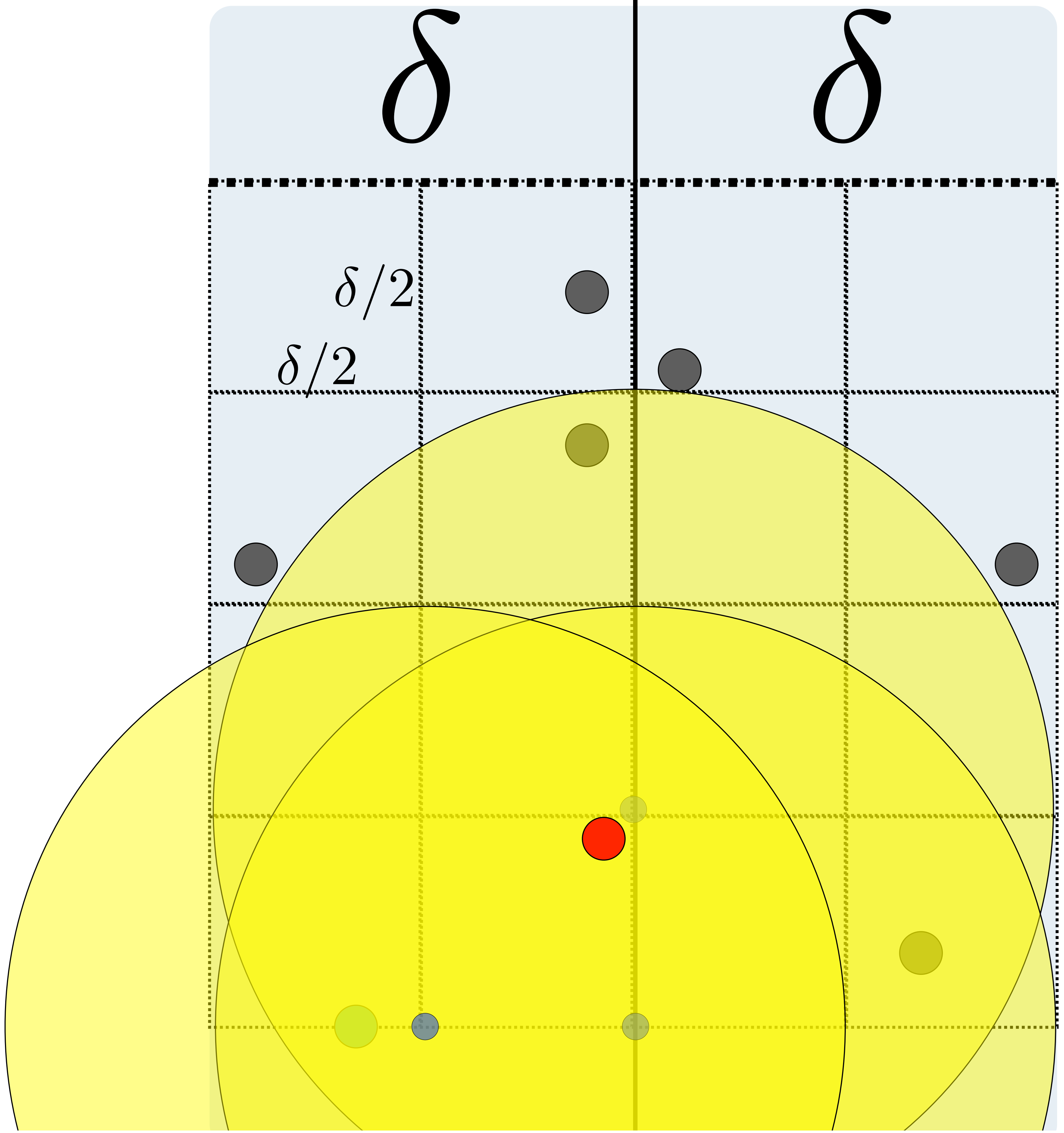

δ
δ
$\delta/2$
$\delta/2$

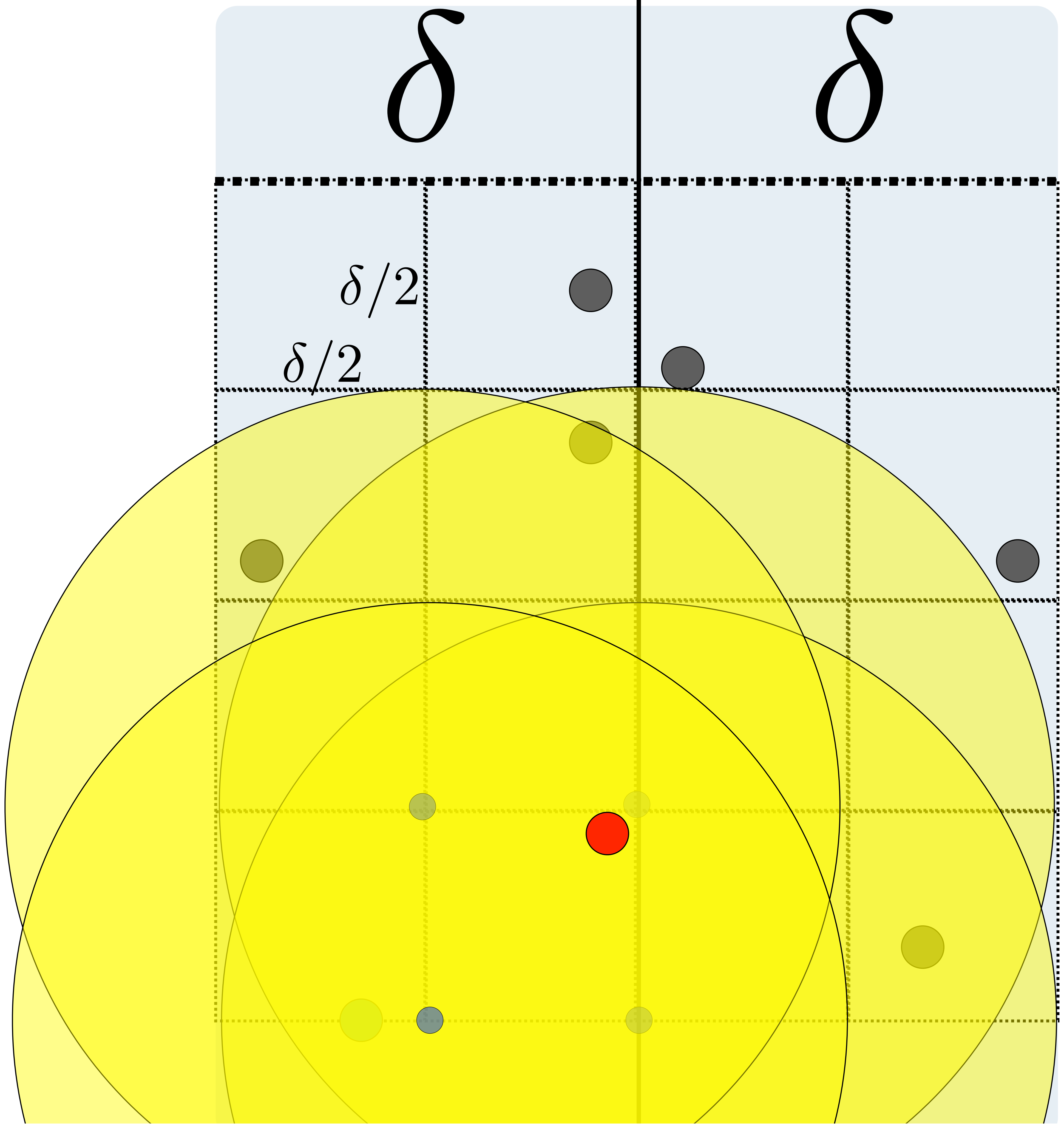

δ
δ
δ/2
δ/2


δ
δ
δ/2
δ/2



δ
δ
δ/2
δ/2

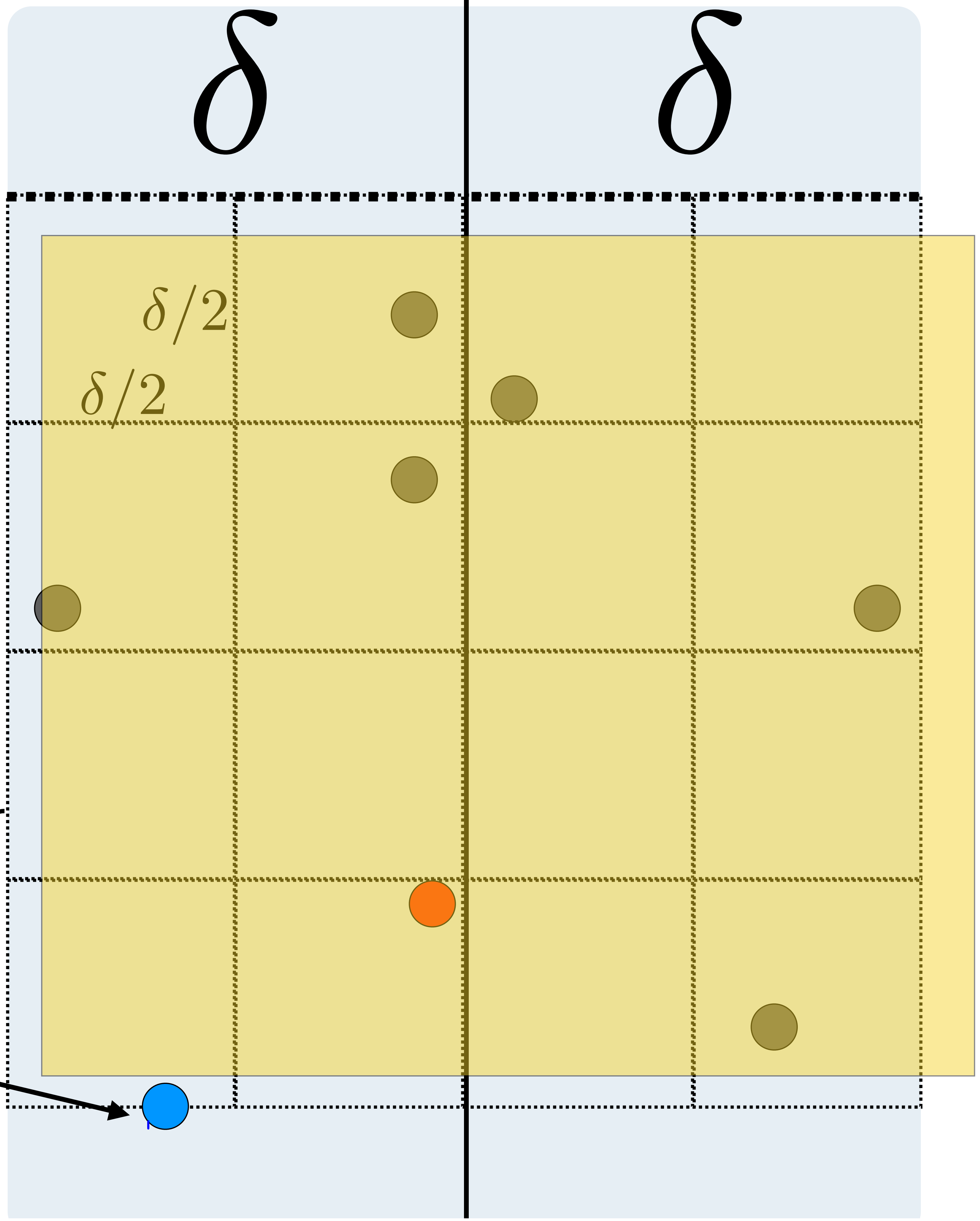

δ
δ
δ/2
δ/2



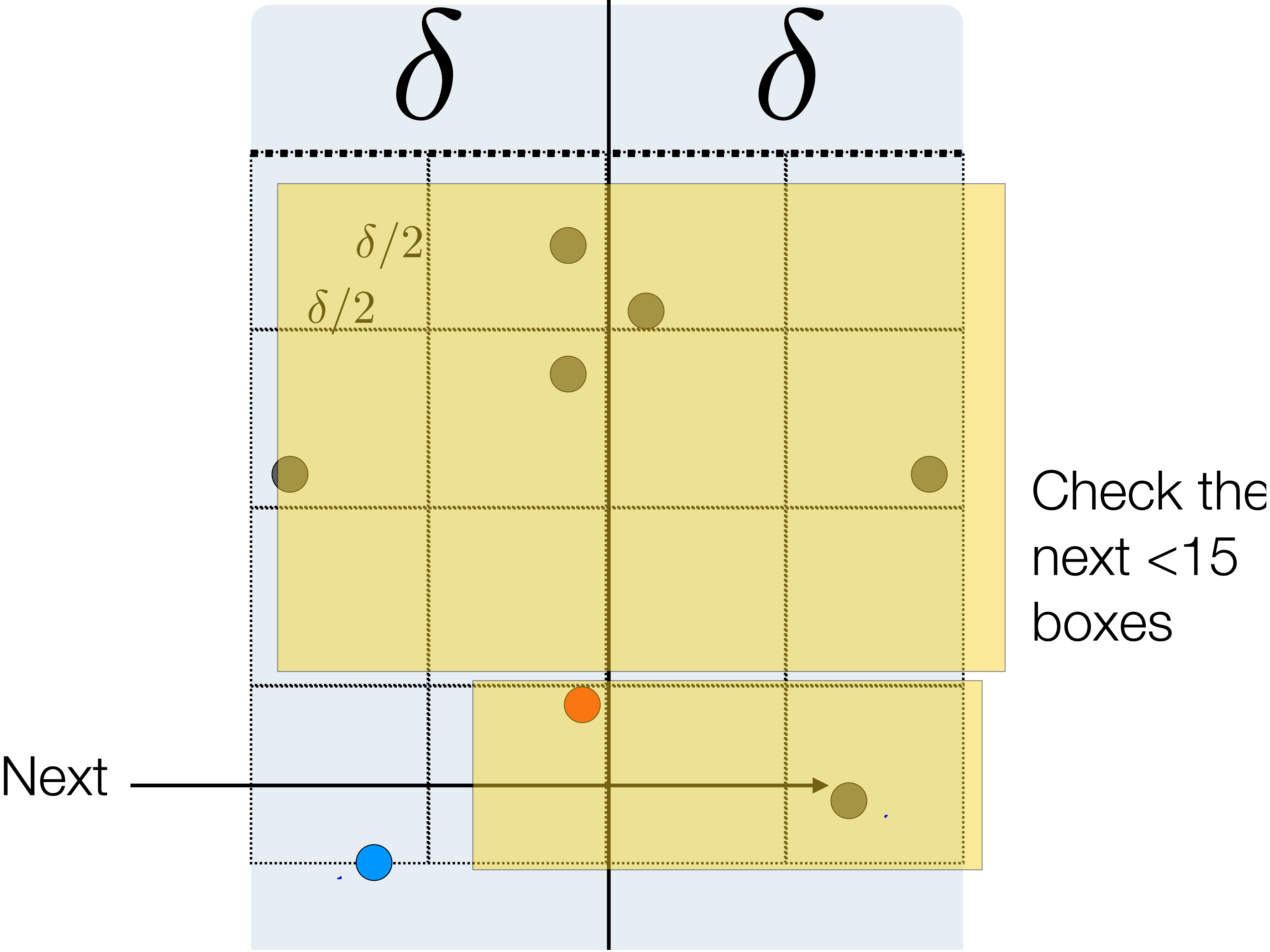
15

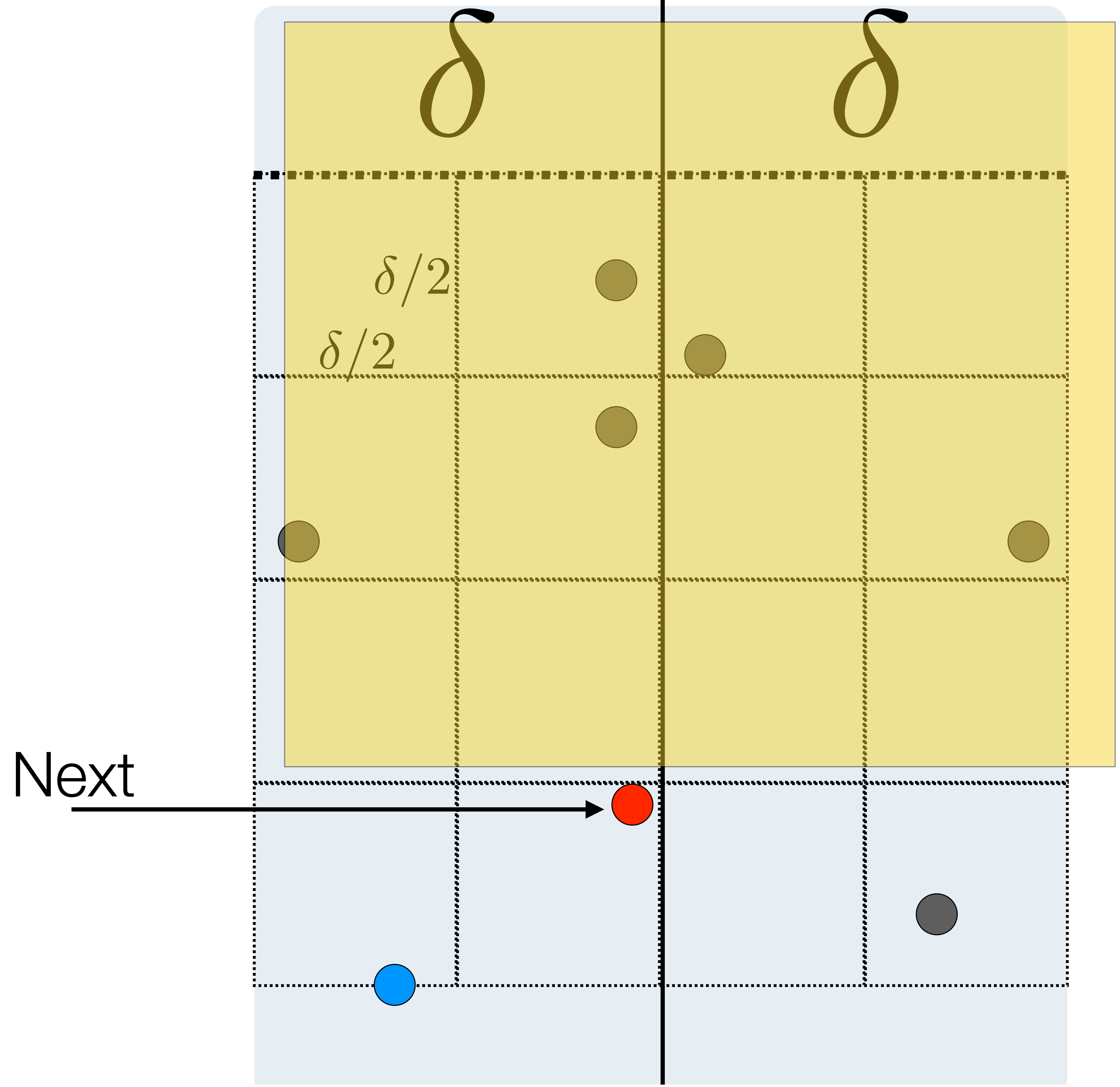
Visit its
by y-order

Start



Check the
next 15
boxes


δ
δ
$\delta/2$
$\delta/2$
Check the
next <15
boxes
Next

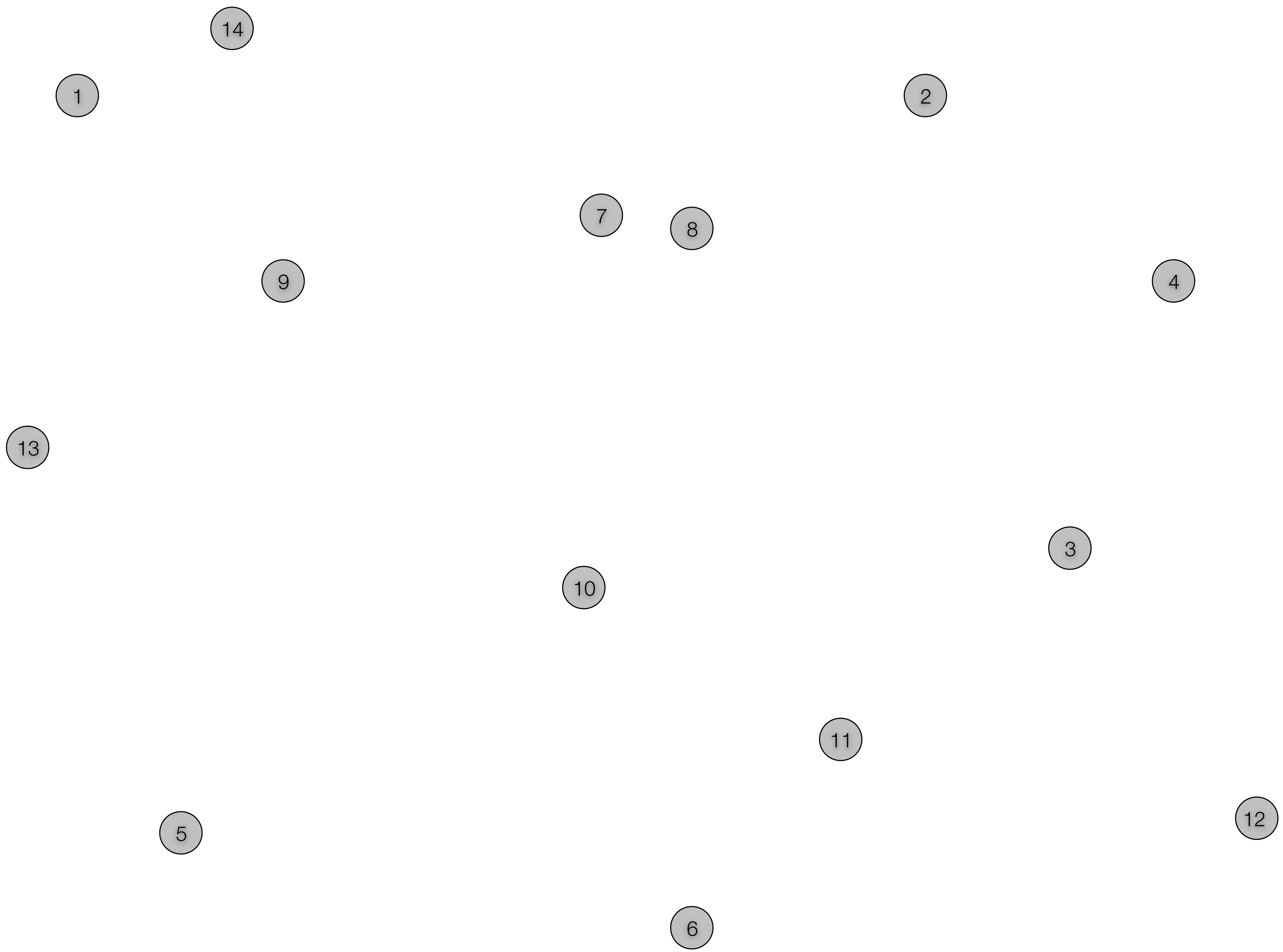


Check the next <15 boxes

```
Closest(P)
    Base Case: If <8 points, brute force.
    Let q be the “middle-element” of points
    Divide P into Left, Right according to q
    delta,r,j = MIN(Closest(Left) ,  Closest(Right) )

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

    For each point x in Mohawk (in order):
        Compute distance to its next 12 neighbors
        Update delta,r,j if any pair (x,y) is < delta

    Return (delta,r,j)
```

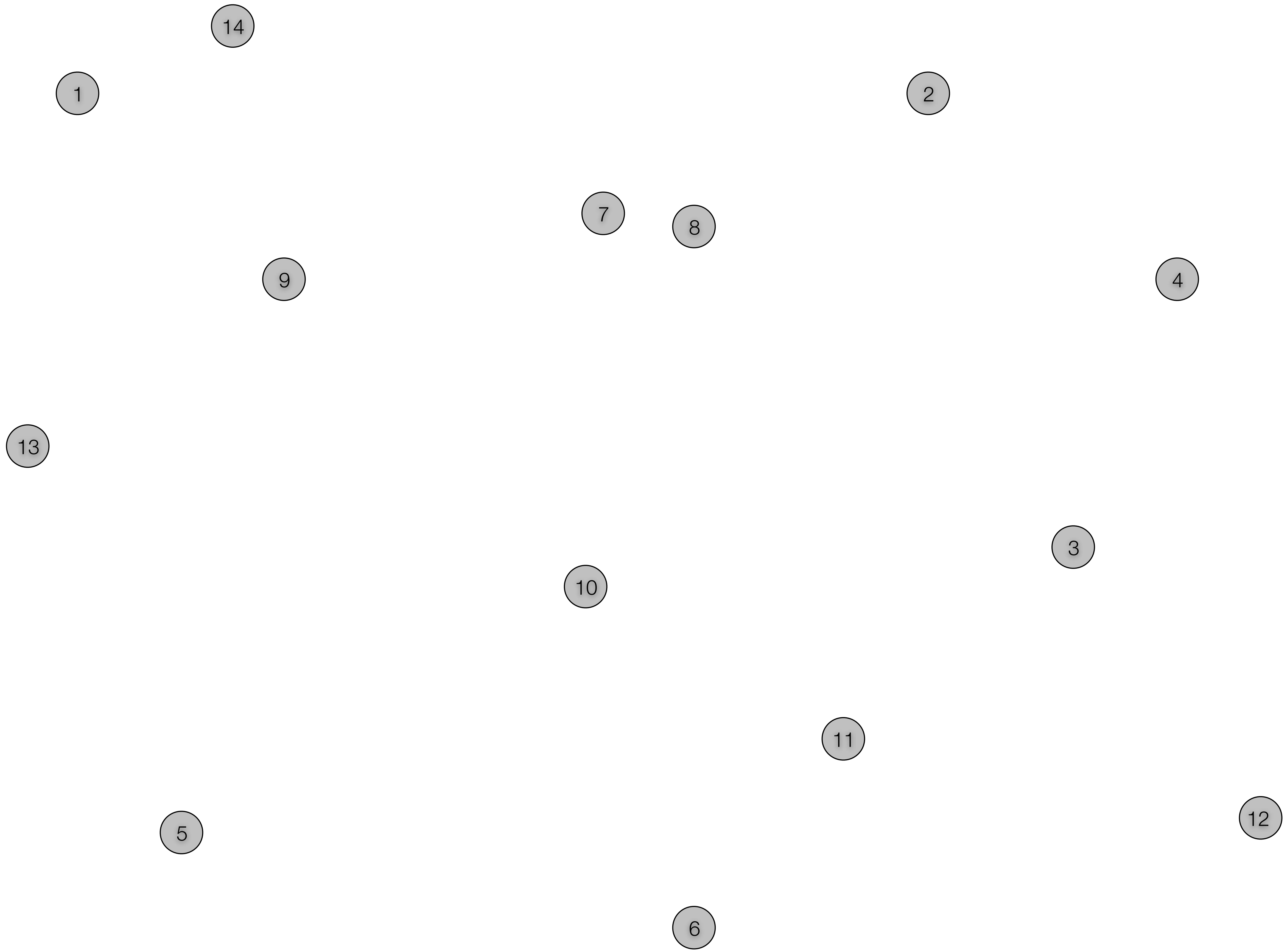
$$T(n) = 2T(n/2) + \Theta(n) = \Theta(n \log n)$$

Details: How to Divide into left/right half ?



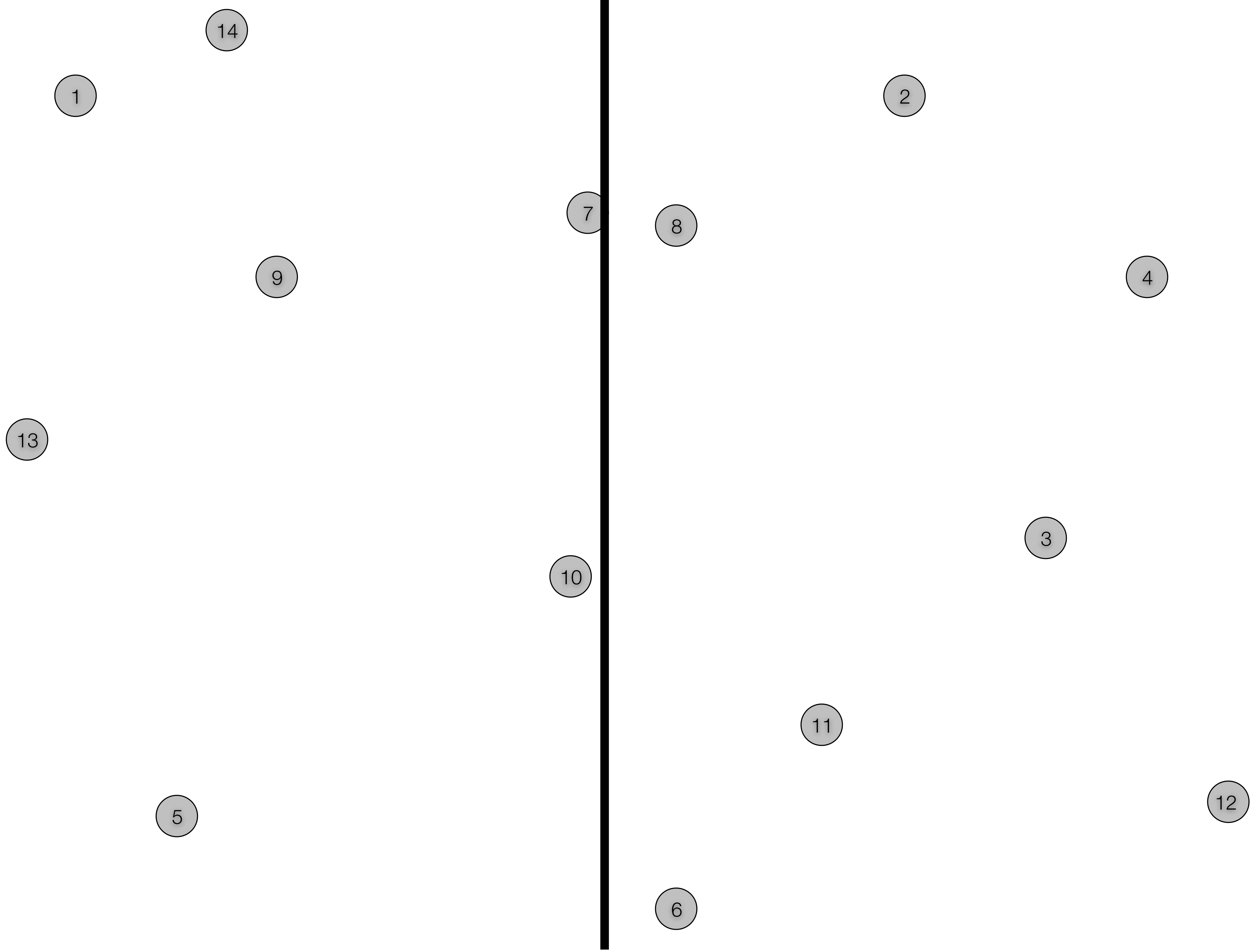
sorted in X: 13 1 5 14 9 10 7 9 8 11 2 3 4 12

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



ClosestPair(P)

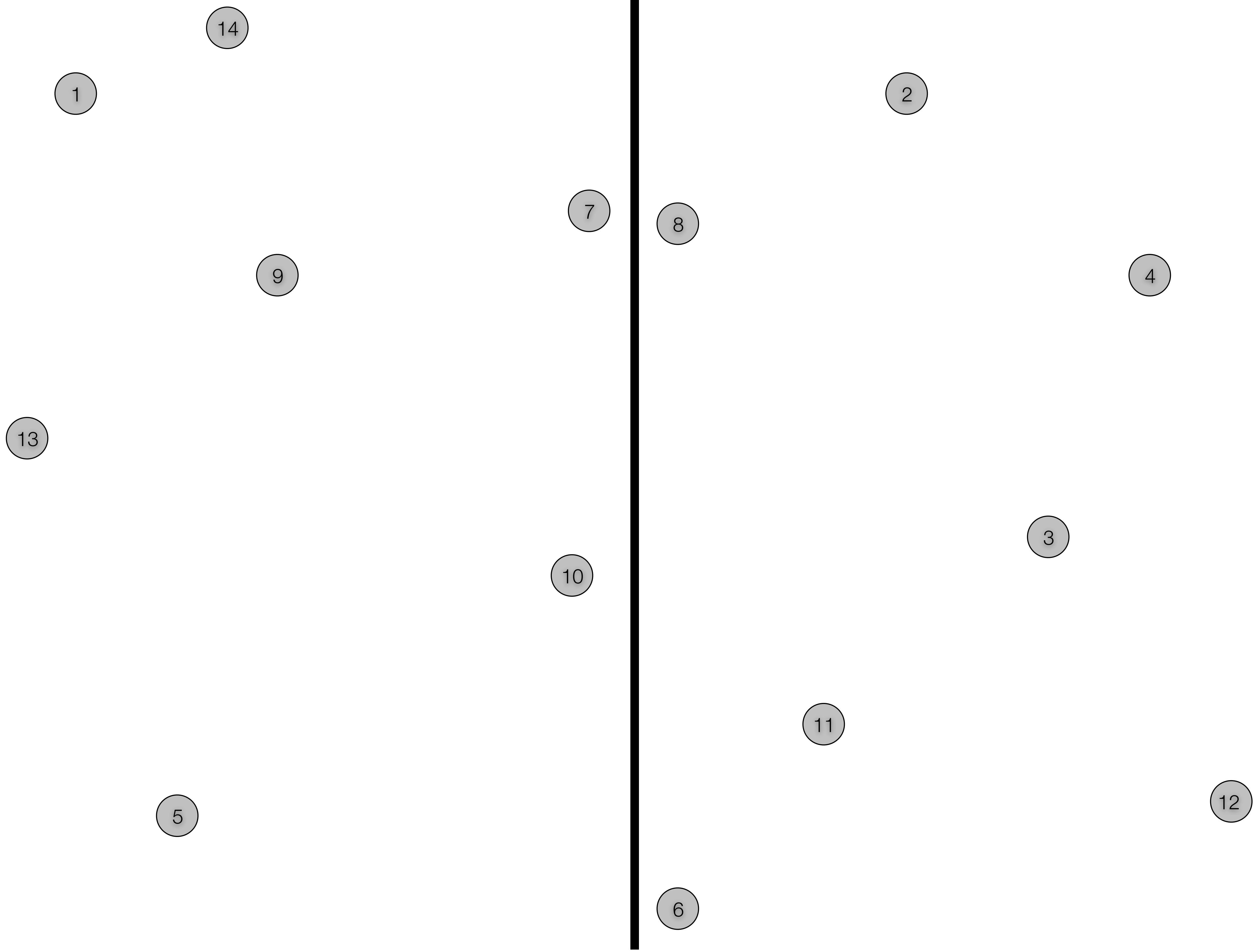
    Compute Sorted-in-X list SX
    Compute Sorted-in-Y list SY
    Closest(P,SX,SY)

Closest(P,SX,SY)

)

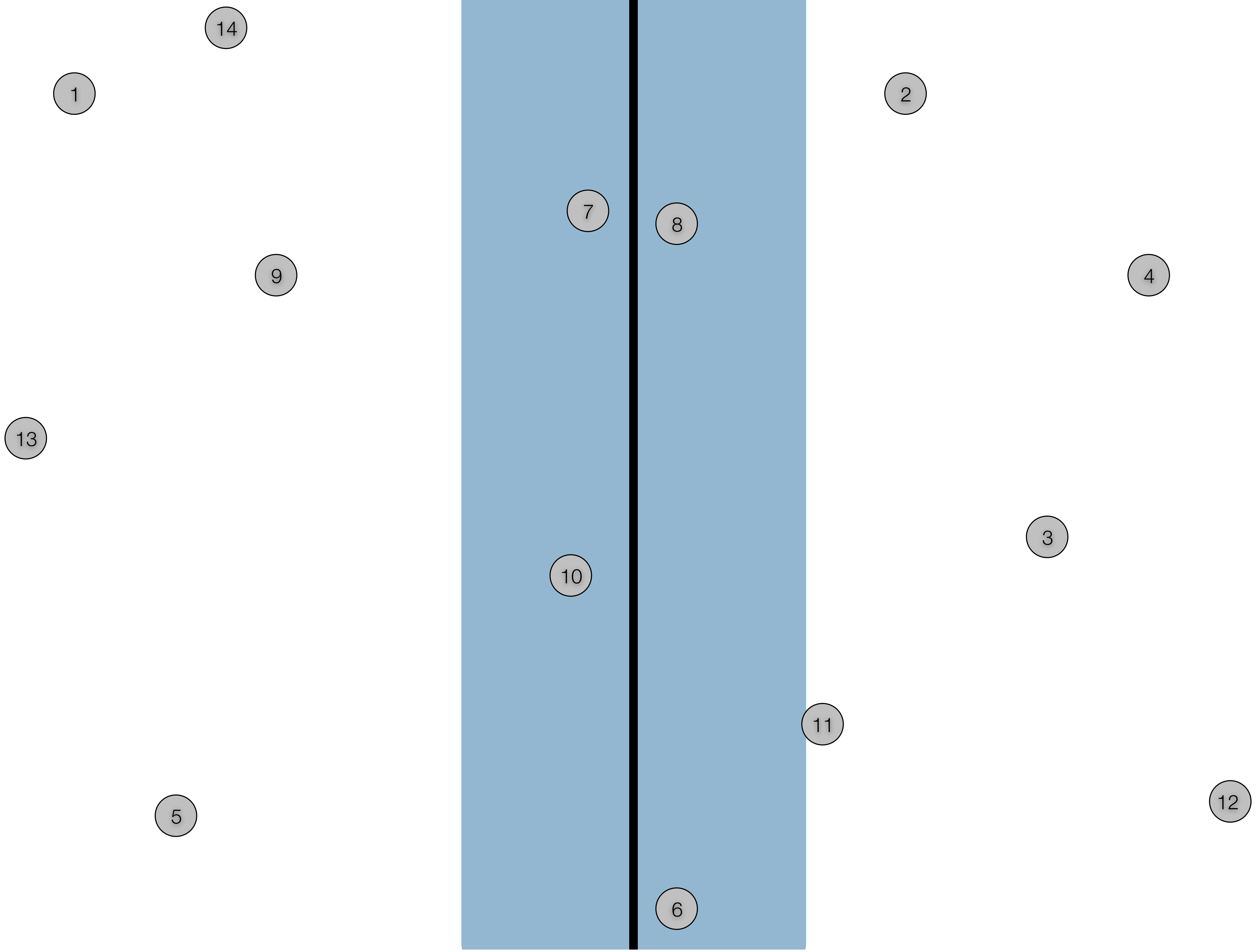
sorted in X: 13 1 5 14 9 10 7 9 8 11 2 3 4 12

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: If <8 points, brute force.

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

Return (delta,r,j)

Closest(P,SX,SY)

Base Case: If <8 points, brute force.

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 12 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

$$T(n) =$$

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