L2 5800

Jan 21/24 2016

Karatsuba, Recurrences

shelat

warmup

Simplify $$(1 + a + a^2 + \cdots + a^L)(a - 1) =$$

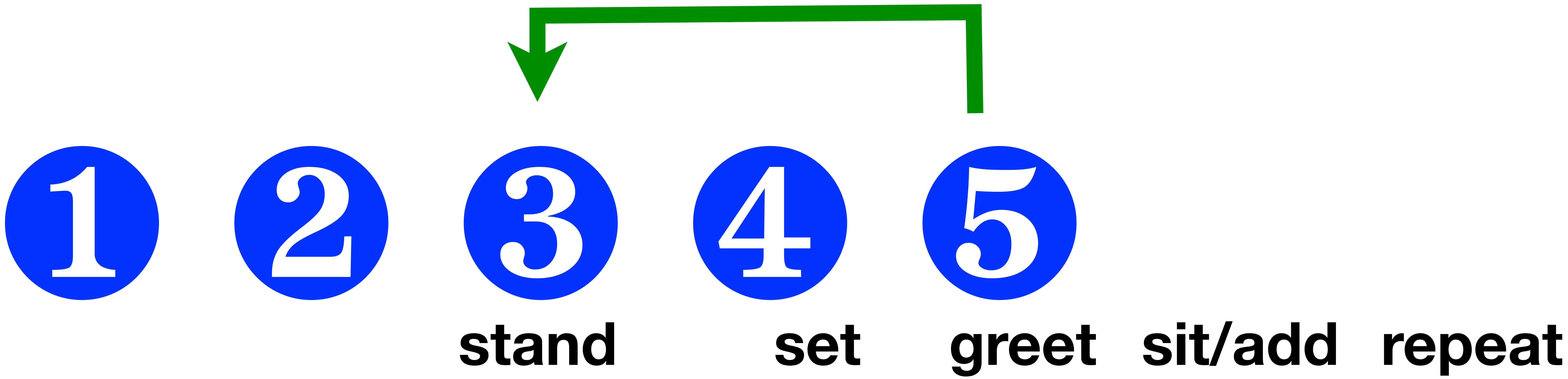
warmup

$$\sum_{i=0}^{L} a^i = \frac{a^{L+1} - 1}{a - 1}$$

Logarithm

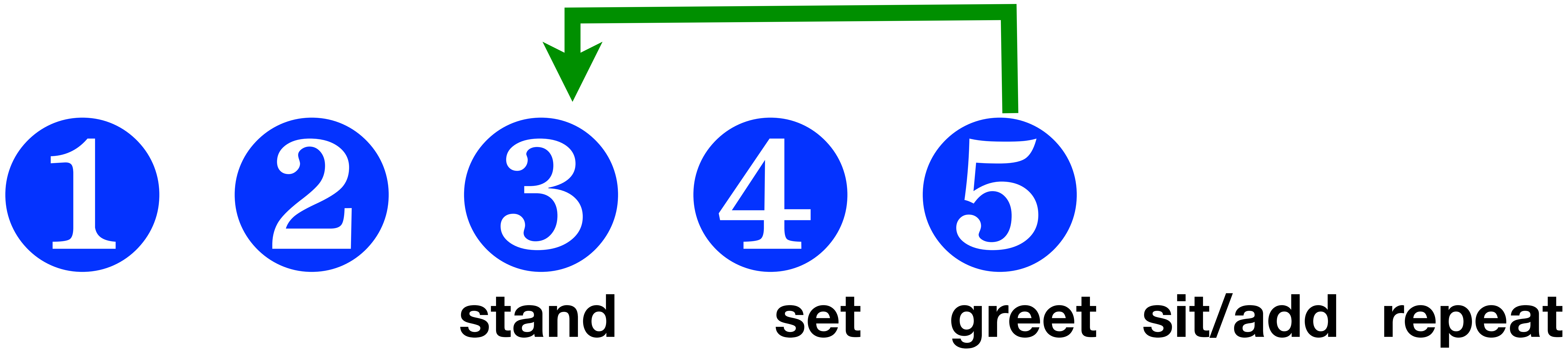
$$\log_2(n) =$$

Recall from last time…

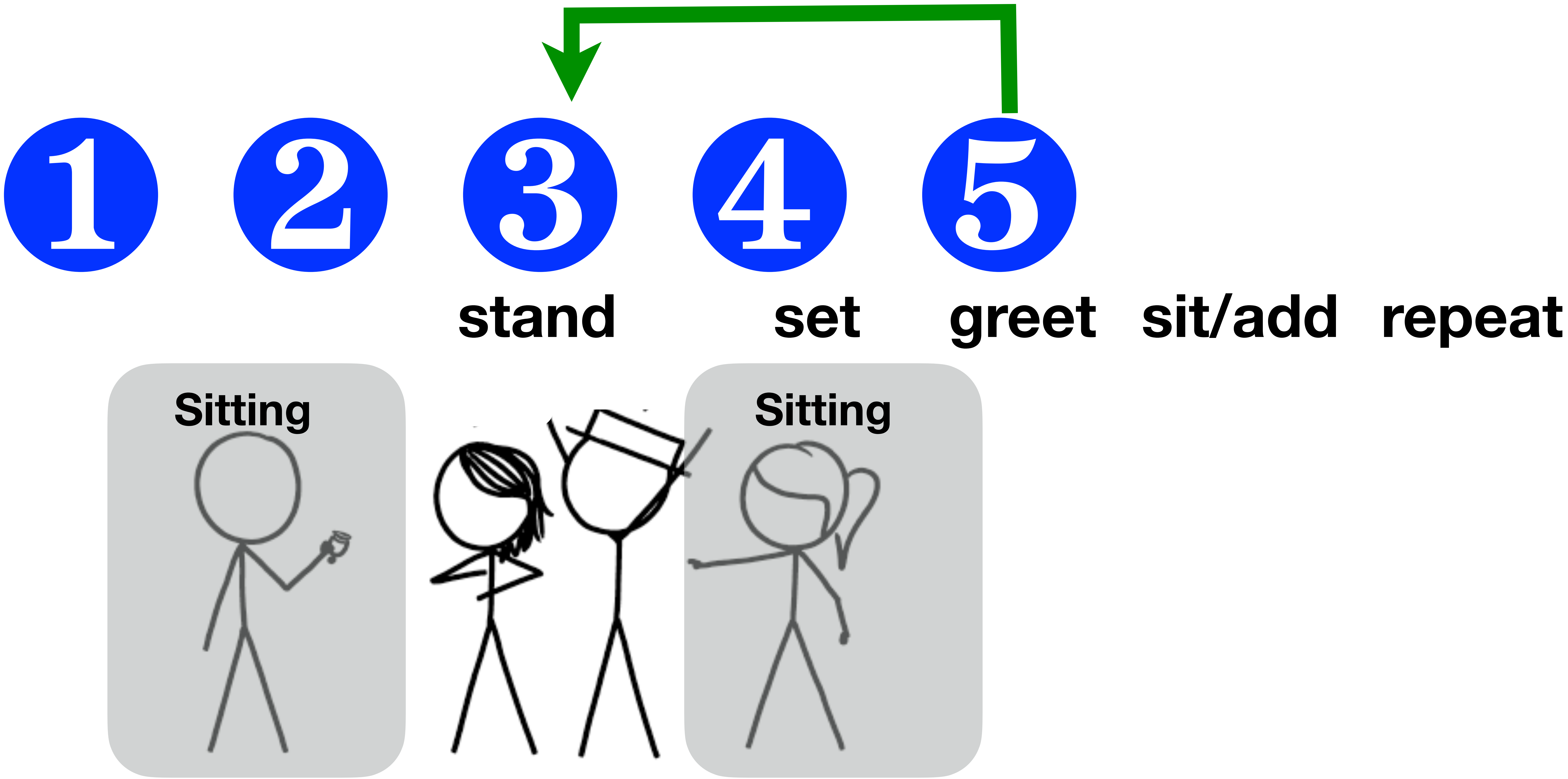


Simple case: 2 people

$$T(2) =$$



$$T(4) =$$



After step 4

$$T(4) =$$

1 stand
2 set

These steps only happen once.

I1:Approx is OK

3 greet
4 sit/add
5 repeat

What about these?

1 stand

2 set

3 greet

4 sit/add

5 repeat

how fast does it work:

$$T(n) = 1 + 1 + T(\lceil n/2 \rceil)$$

1 stand
2 set
3 greet
4 sit/add
5 repeat

how fast does it work:

$$T(n) = 1 + 1 + T(\lceil n/2 \rceil)$$

$$T(1) = 3$$

This is a recurrence

$$T(n) = T(\lceil n/2 \rceil) + 2$$

$$T(1) = 3$$

solve a simpler case when n is a power of 2.

$$T(2^k) = 2 + T(2^{k-1})$$

$$T(2^k) = 2 + T(2^{k-1})$$

$$T(2^k) = 2 + T(2^{k-1})$$

$$= 2 + 2 + T(2^{k-2})$$

$$T(2^k) = 2 + T(2^{k-1})$$

$$= 2 + 2 + T(2^{k-2})$$

$$= \overbrace{2 + 2 + \cdots + 2}^{k} + T(2^0)$$

$$T(2^k) = 2 + T(2^{k-1})$$

$$= 2 + 2 + T(2^{k-2})$$

$$= \overbrace{2 + 2 + \cdots + 2}^{k} + T(2^0)$$

$$= 2k + T(1)$$

intuition here

$$T(2^k) = 2 + T(2^{k-1})$$

Other cases?

intuition here

$$T(2^k) = 2 + T(2^{k-1})$$

$$= 2 + 2 + T(2^{k-2})$$

Other cases?

intuition here

$$T(2^k) = 2 + T(2^{k-1})$$

$$= 2 + 2 + T(2^{k-2})$$

$$= \overbrace{2 + 2 + \cdots + 2}^{k} + T(2^0)$$

Other cases?

intuition here

$$T(2^k) = 2 + T(2^{k-1})$$

$$= 2 + 2 + T(2^{k-2})$$

$$= \overbrace{2 + 2 + \cdots + 2}^{k} + T(2^0)$$

$$= 2k + T(1)$$

Other cases?

Idea1: It is OK to approximate

A good way to do this is to ignore low order terms of our functions, i.e. using asymptotic notation for our functions.

Asymptotic notation

$O(g)$ This notation represents a set

Asymptotic notation

$$O(g)$$

Set of functions that are at most within constant of g for large n

Asymptotic notation

$$O(g)$$

Set of functions that are at *most* within const of g for large n

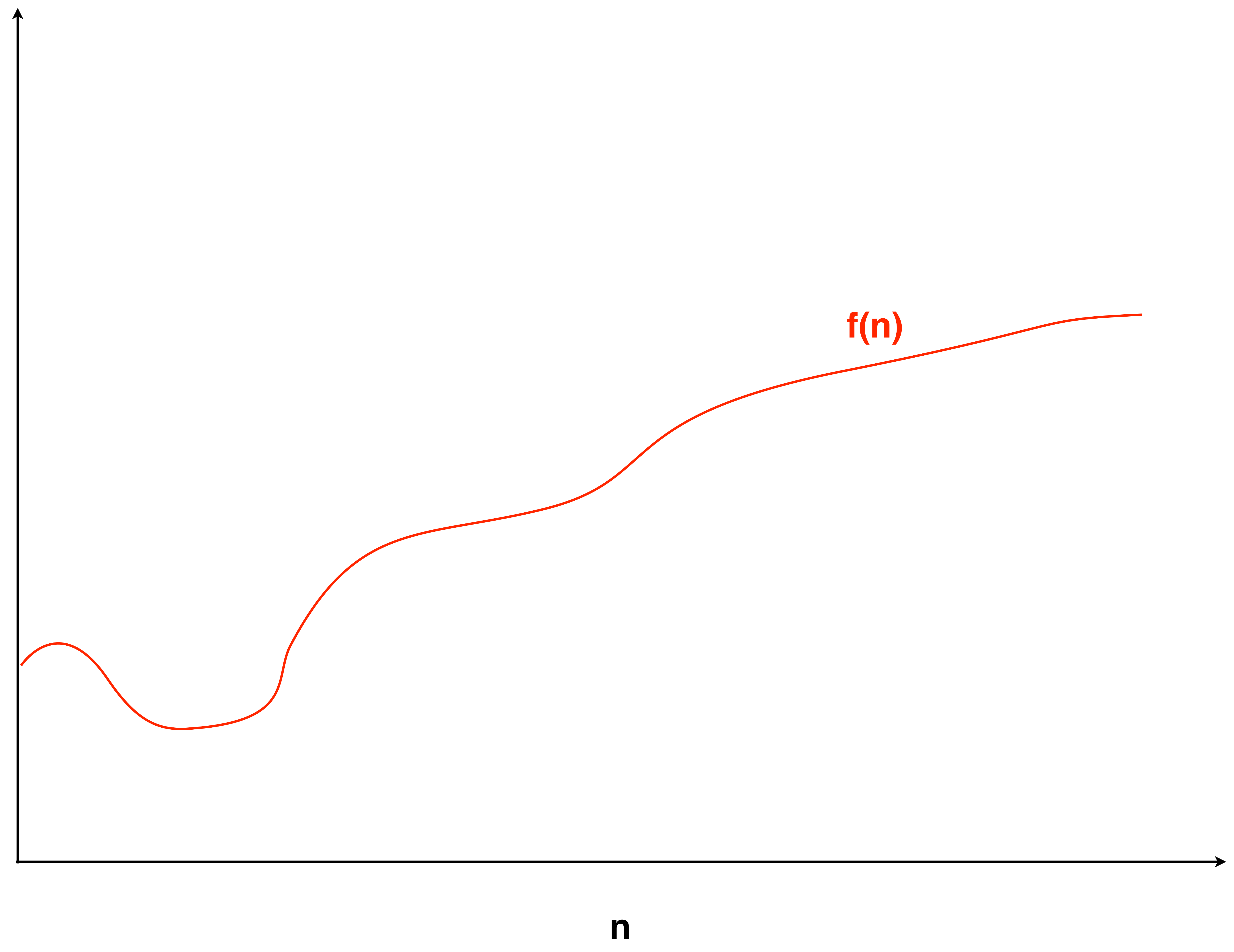
$$\Omega(g)$$

Set of functions that are at *least* within const of g for large n

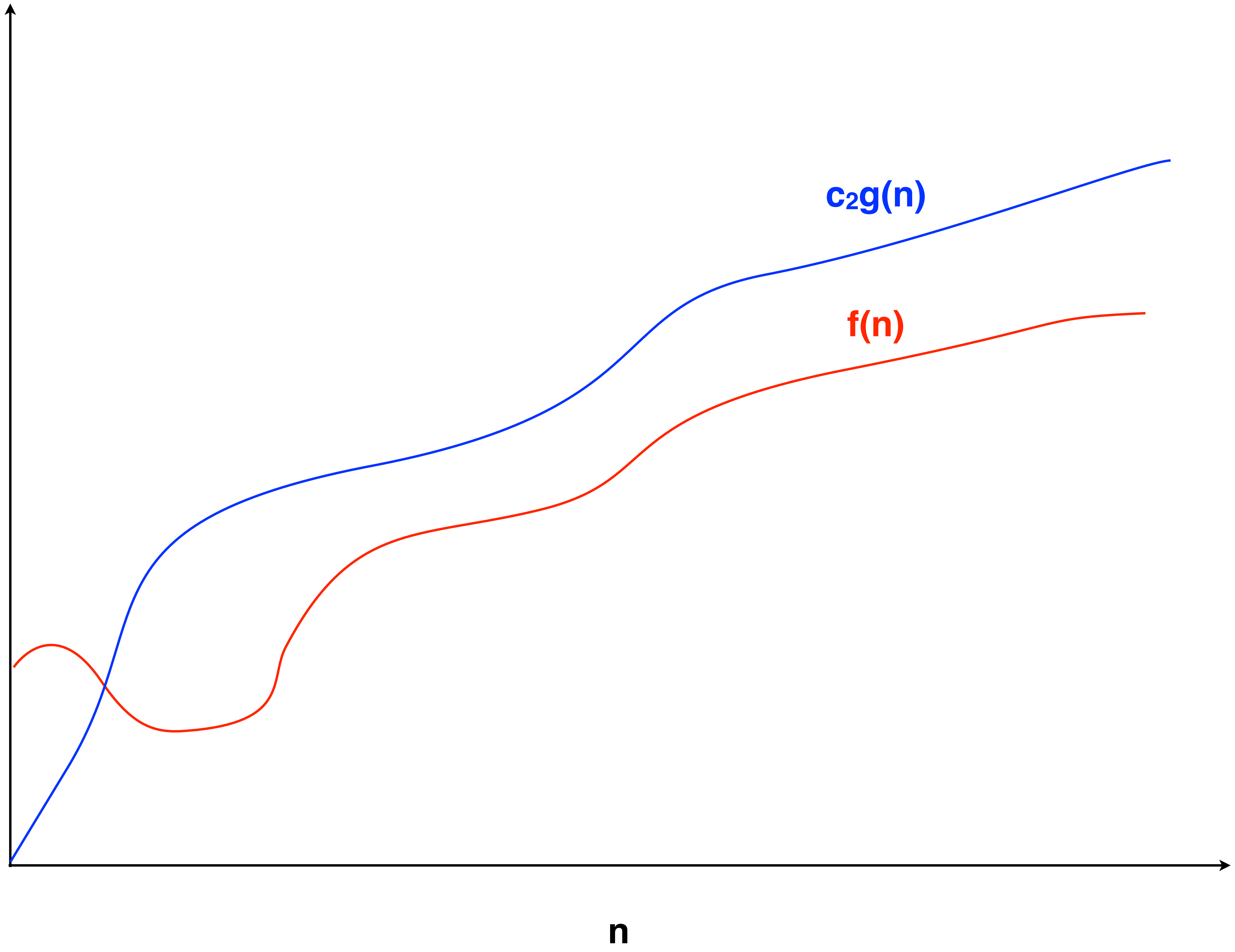
$$\Theta(g)$$

Set of functions that are at within const of g for large n

Omega sandwich

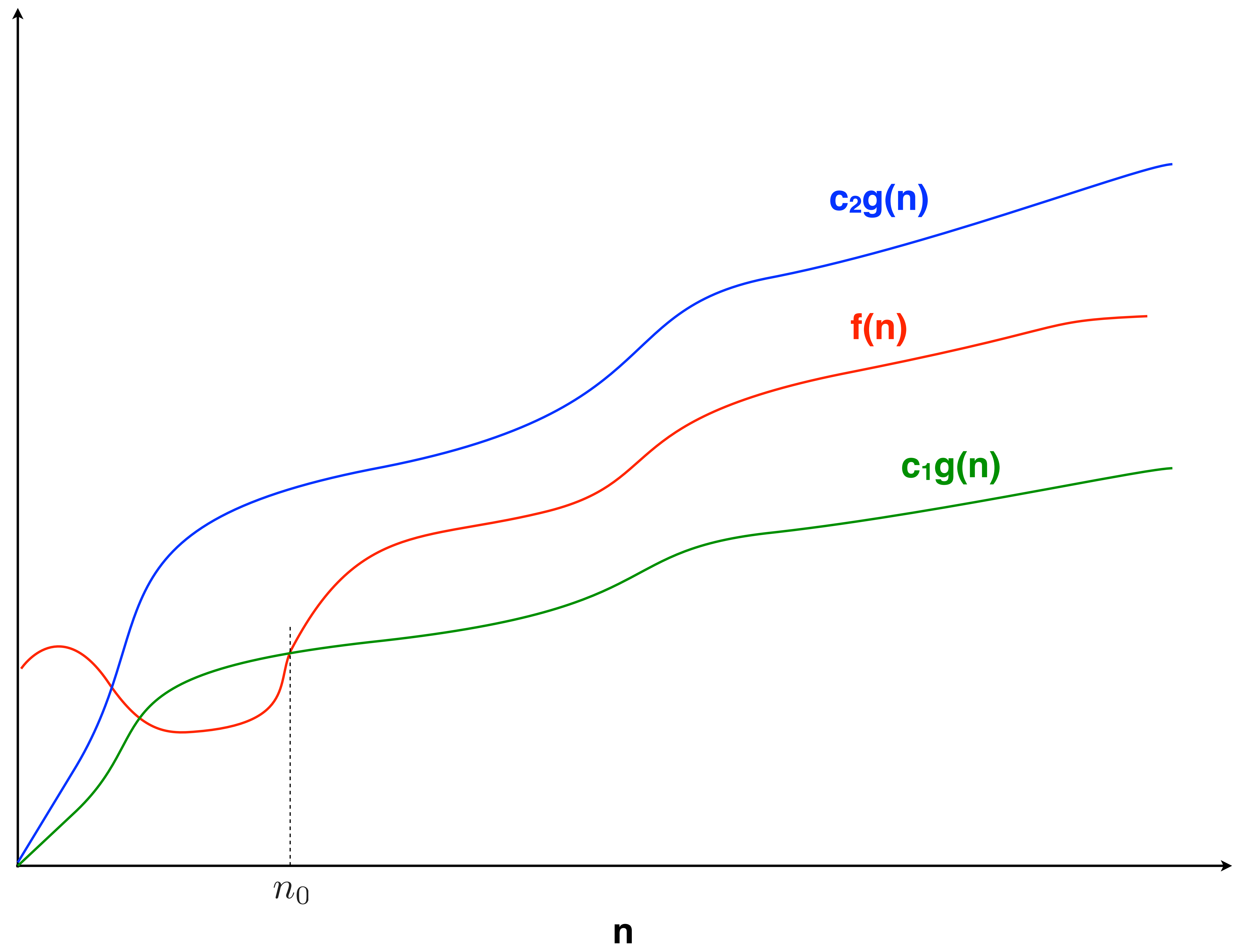


Omega sandwich



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

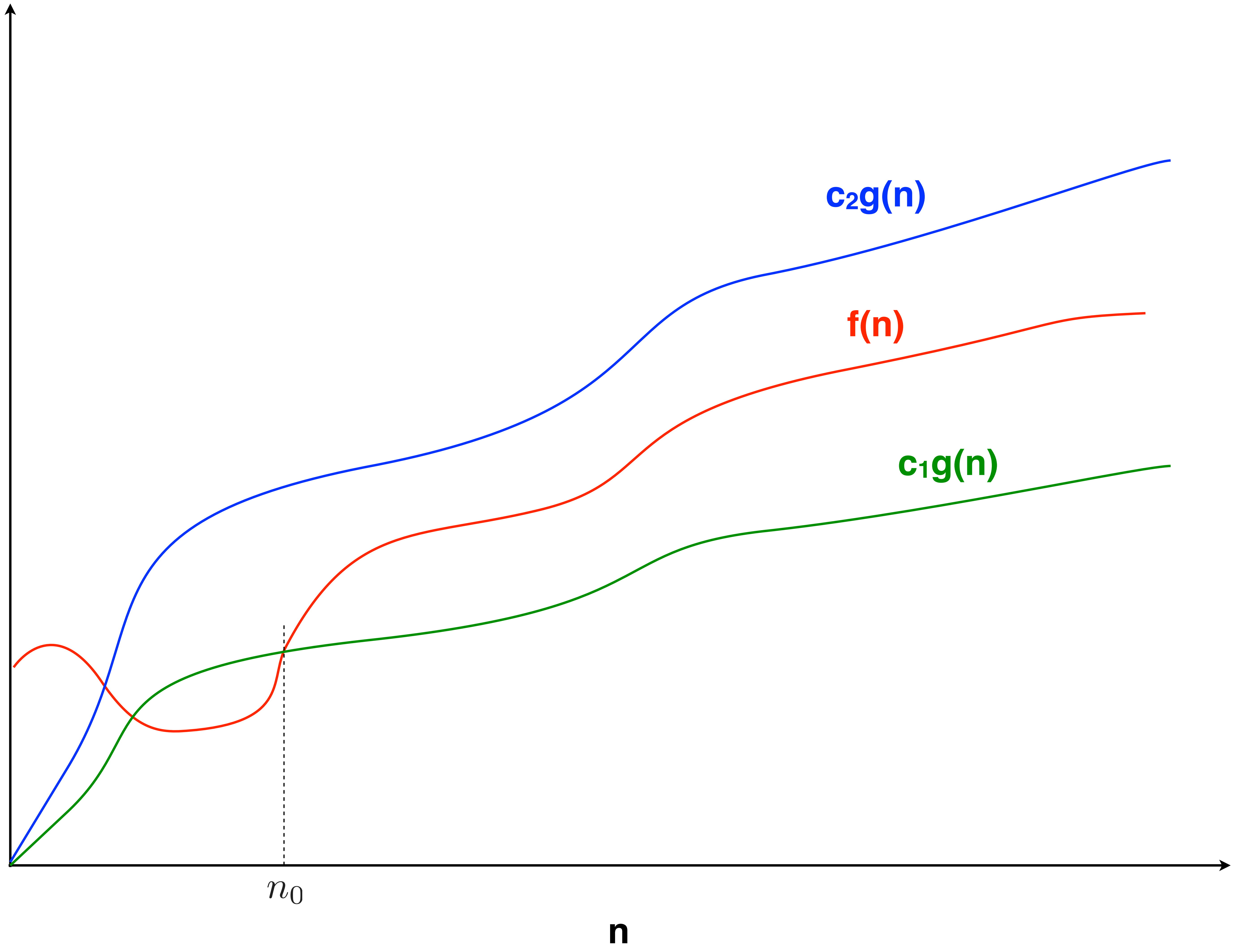
Omega sandwich



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

$$f(n) = \Omega(g(n))$$

Omega sandwich



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

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

$$f(n) = \Omega(g(n))$$

Examples of asymptotic notation

intuition here

$$T(2^k) = 2 + T(2^{k-1})$$

$$= 2 + 2 + T(2^{k-2})$$

$$= \overbrace{2 + 2 + \cdots + 2}^{k} + T(2^0)$$

$$= 2k + T(1) \;= O(\log(2^k))$$

intuition here

$$T(2^k) = 2 + T(2^{k-1})$$

$$= 2 + 2 + T(2^{k-2})$$

$$= \overbrace{2 + 2 + \cdots + 2}^{k} + T(2^0)$$

$$= 2k + T(1) \;= O(\log(2^k))$$

$$\forall 0 < n < m, T(n) \leq T(m)$$

$$T(m) \leq T(2^{\lceil \log(m) \rceil}) = 2\lceil \log(m) \rceil + 2$$

intuition here

$$T(2^k) = 2 + T(2^{k-1})$$

$$= 2 + 2 + T(2^{k-2})$$

$$= \overbrace{2 + 2 + \cdots + 2}^{k} + T(2^0)$$

$$= 2k + T(1) \; = O(\log(2^k))$$

$$\forall 0 < n < m, T(n) \leq T(m)$$

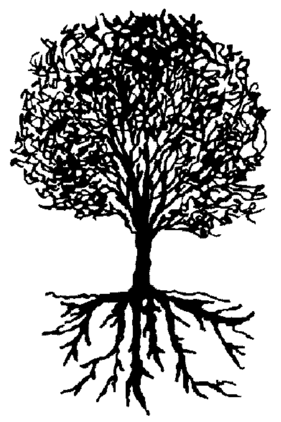
$$T(m) \leq T(2^{\lceil \log(m) \rceil}) = 2\lceil \log(m) \rceil + 2$$

$$T(m) = \Omega(\log(m))$$

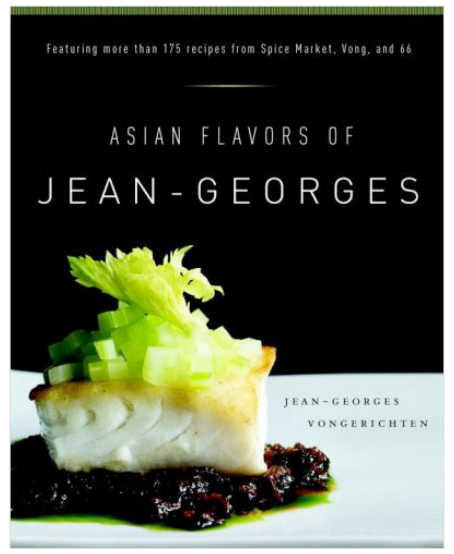
$$= \Theta(\log(m))$$

main ideas:

How to solve recurrence relations



?-✓

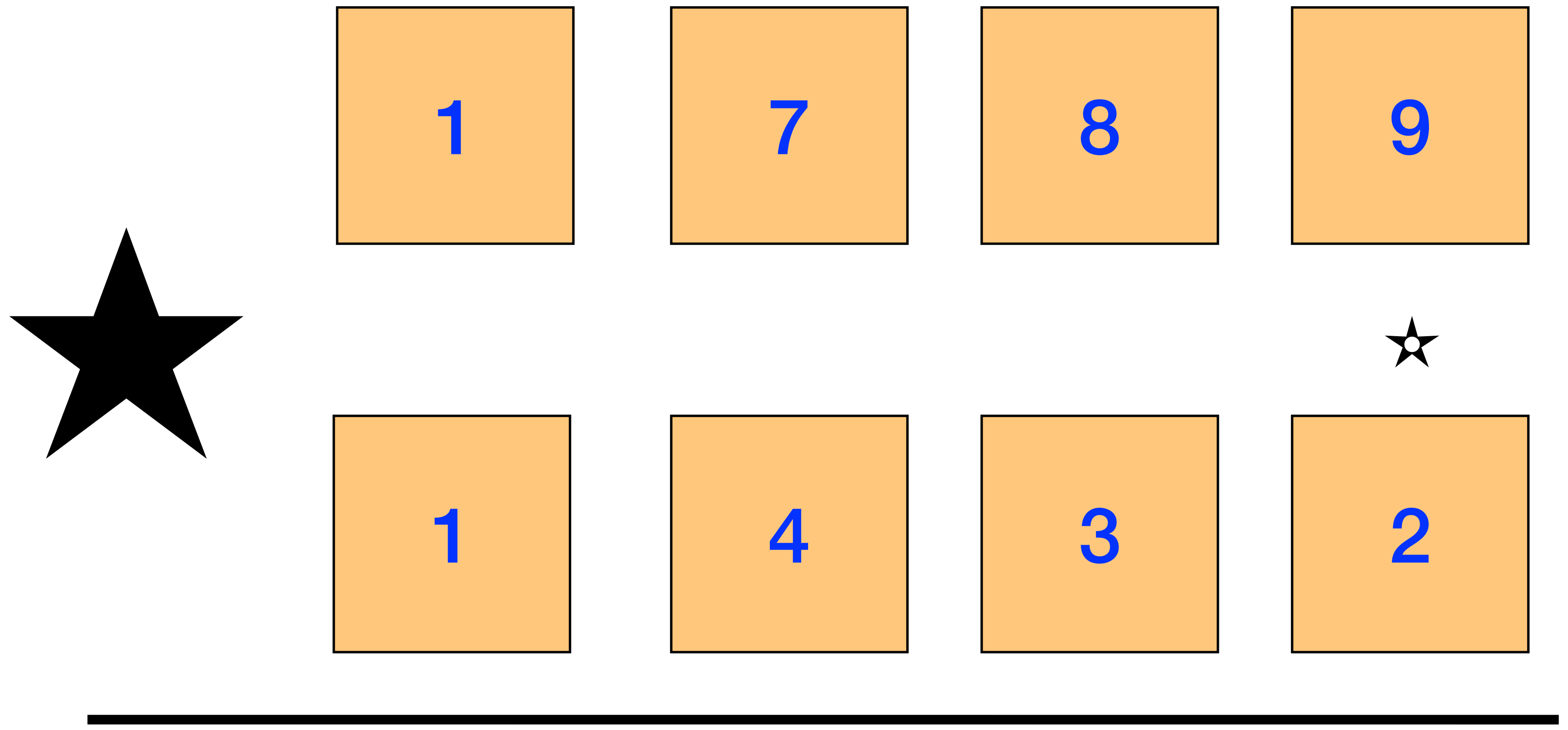


http://www.drblank.com/law301.jpg

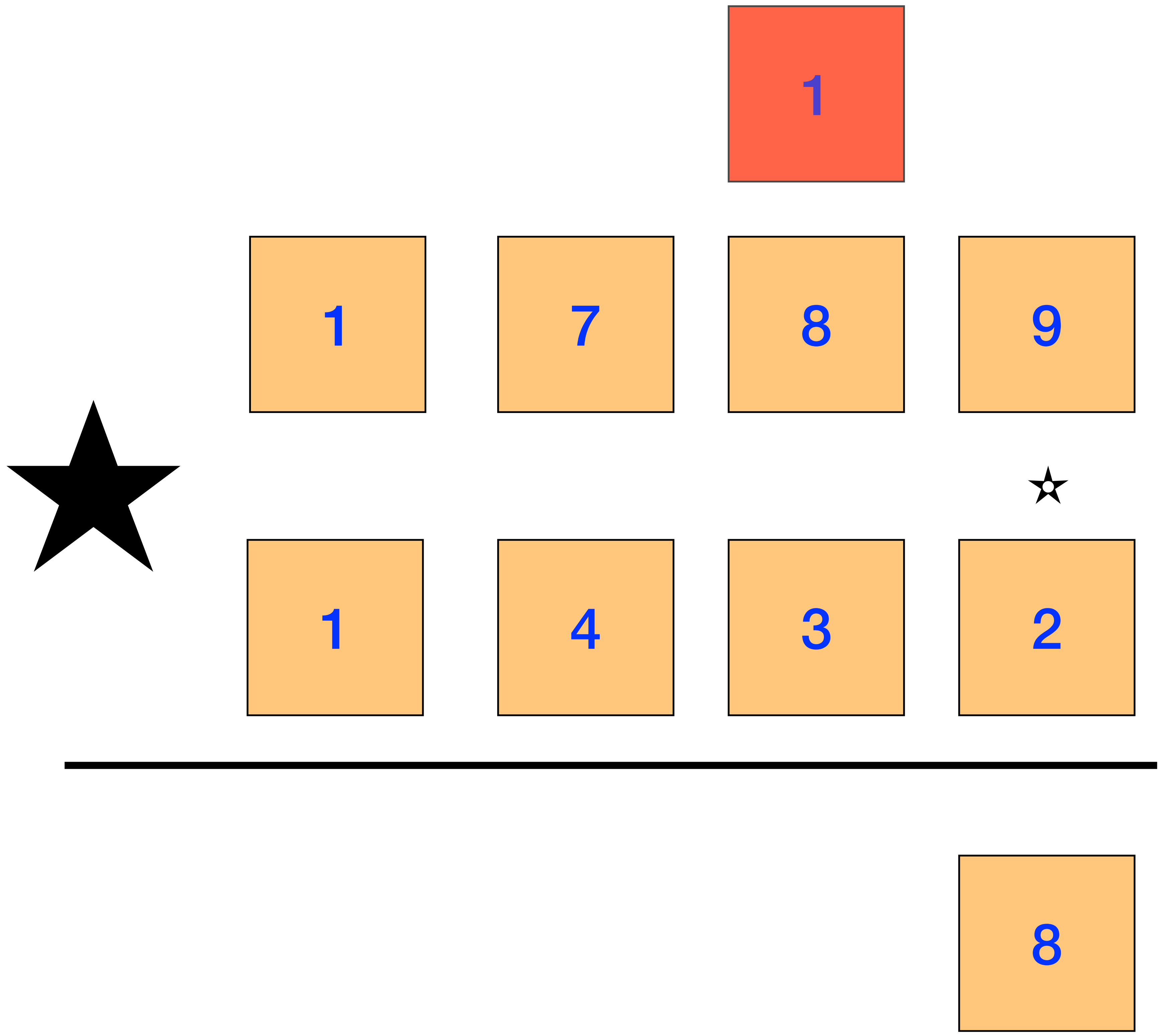
Multiplication

	1	7	8	9
★	1	4	3	2

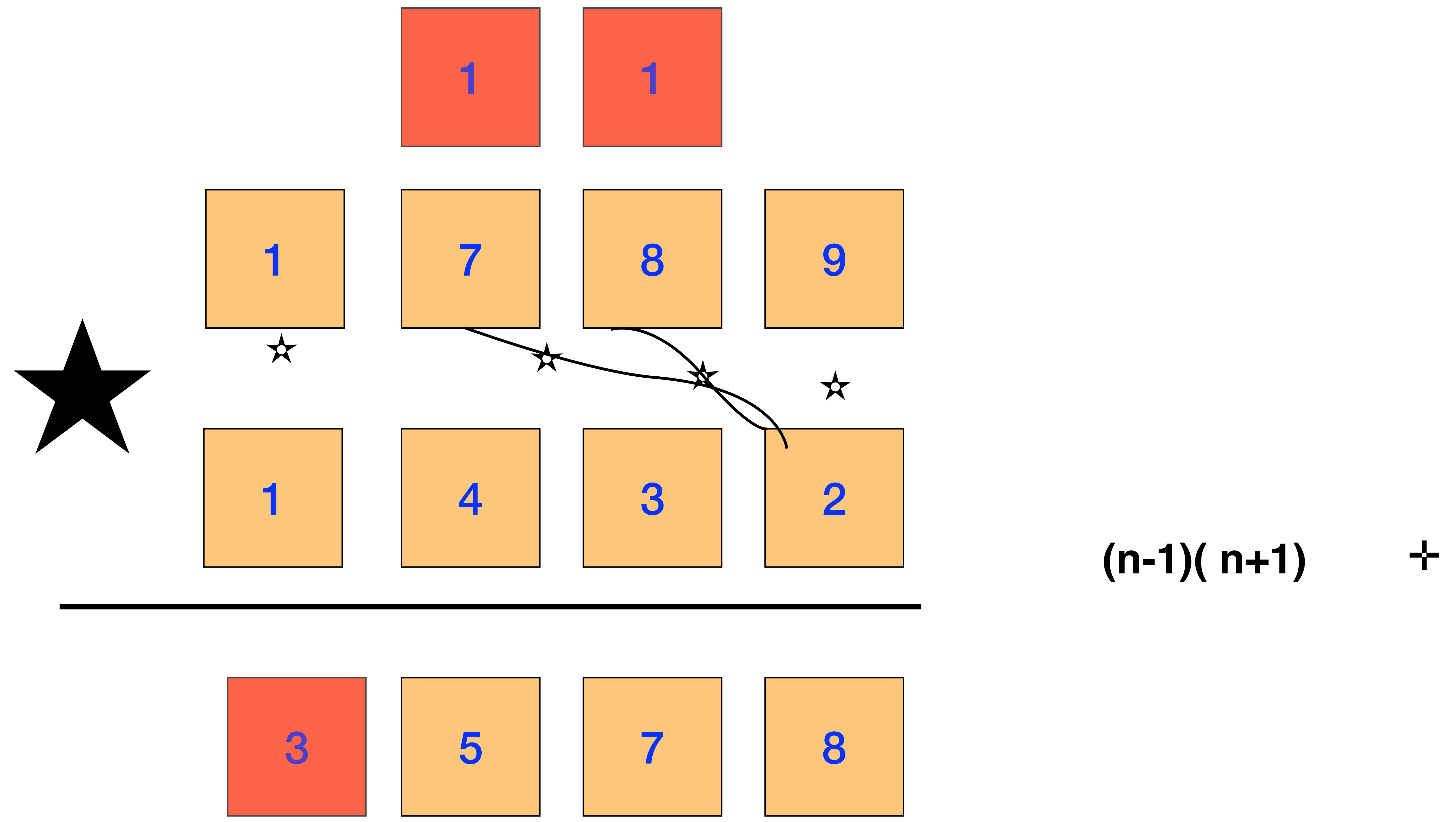
(n-1)(n+1) ✛



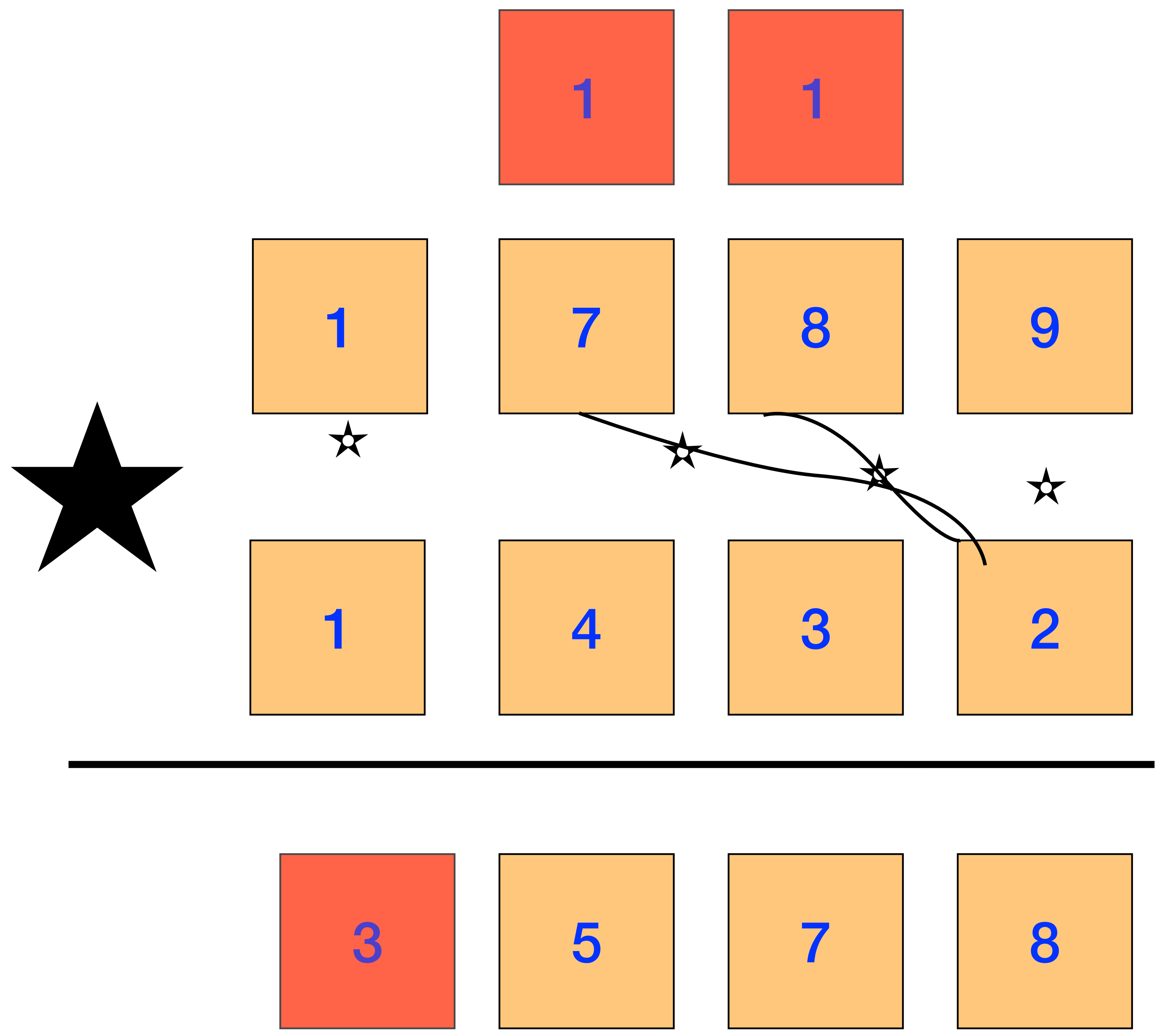
$(n-1)(n+1)$



(n-1)(n+1)

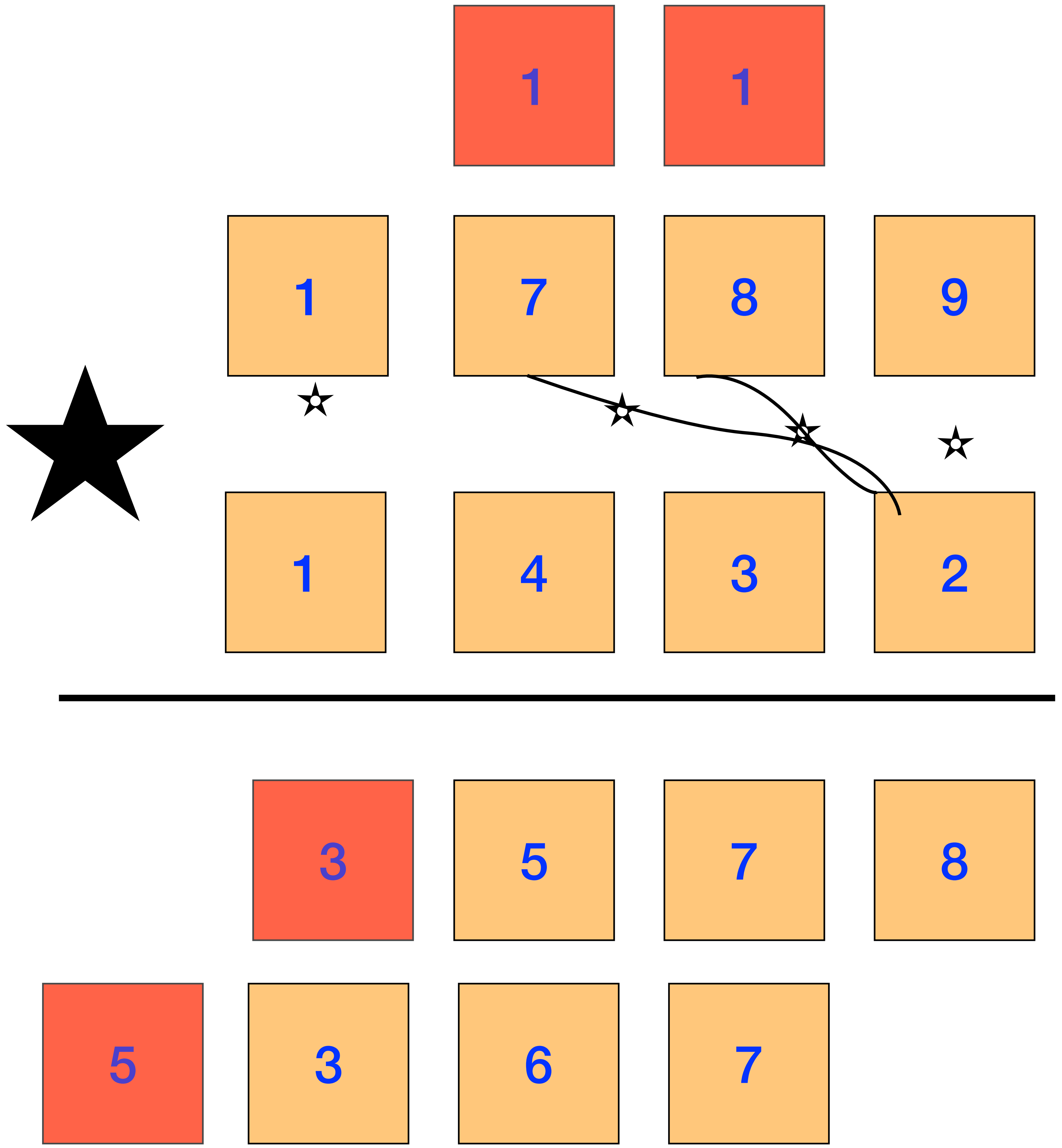


1 1
1 7 8 9
1 4 3 2
3 5 7 8
(n-1)(n+1)



(n-1)(n+1) ✢

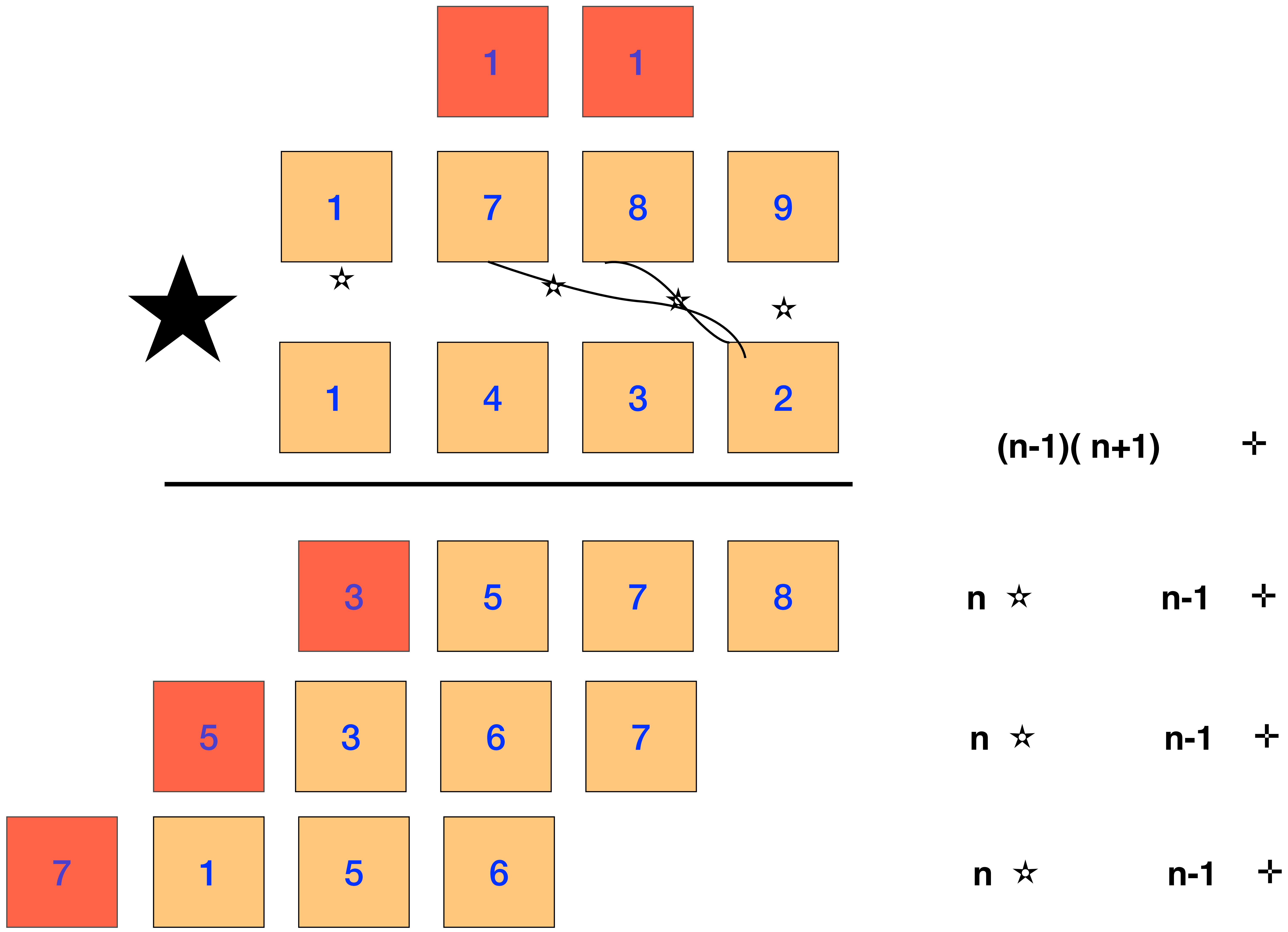
n ☆ n-1 ✢



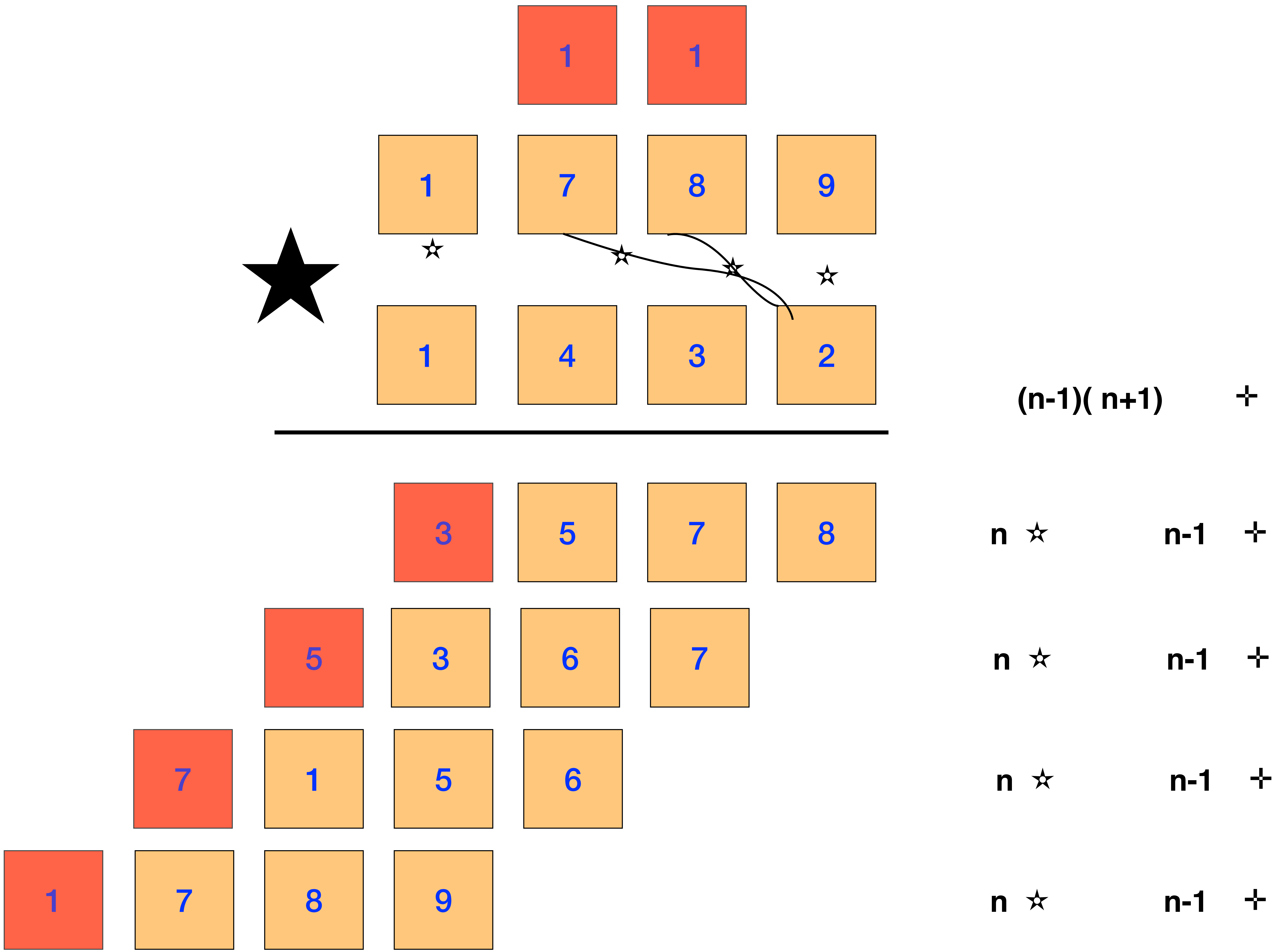
(n-1)(n+1) ✢

n ☆ n-1 ✢

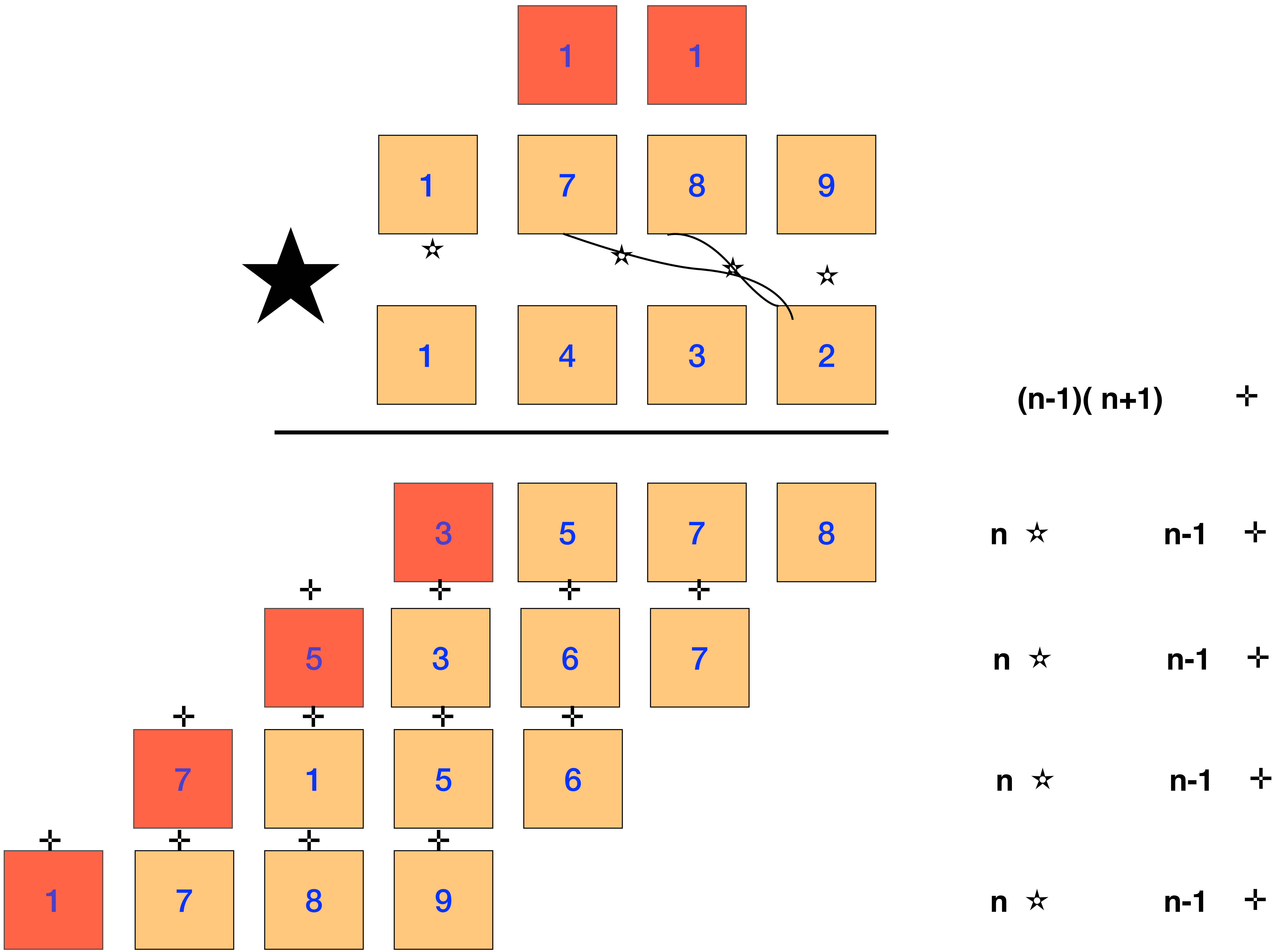
n ☆ n-1 ✢



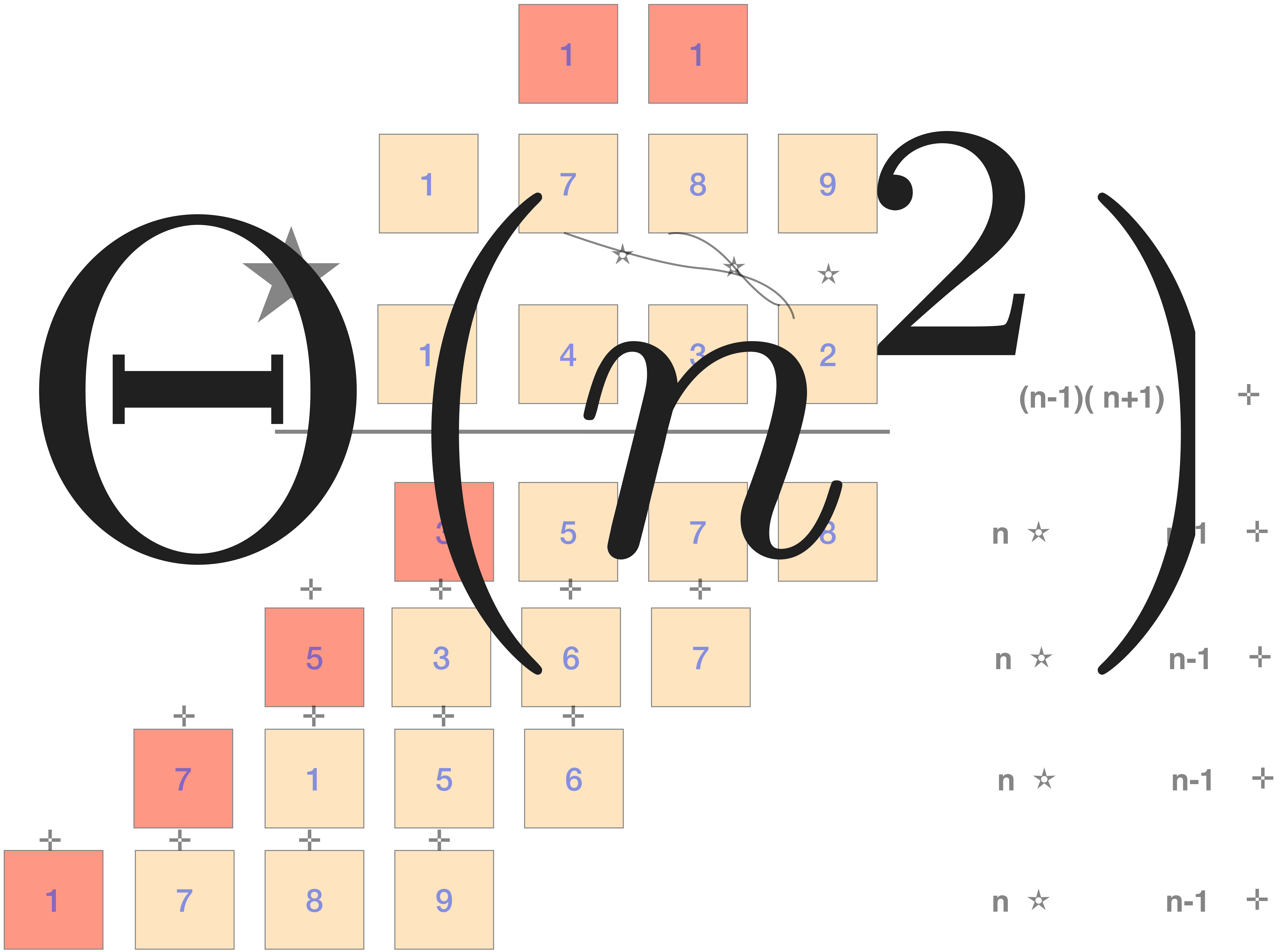
1 1
1 7 8 9
1 4 3 2
(n-1)(n+1) ✢
3 5 7 8
n ☆ n-1 ✢
5 3 6 7
n ☆ n-1 ✢
7 1 5 6
n ☆ n-1 ✢



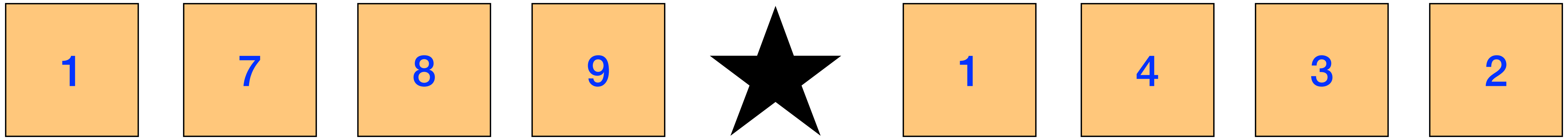
1 1
1 7 8 9
1 4 3 2
(n-1)(n+1) ✢
3 5 7 8
n ☆ n-1 ✢
5 3 6 7
n ☆ n-1 ✢
7 1 5 6
n ☆ n-1 ✢
1 7 8 9
n ☆ n-1 ✢

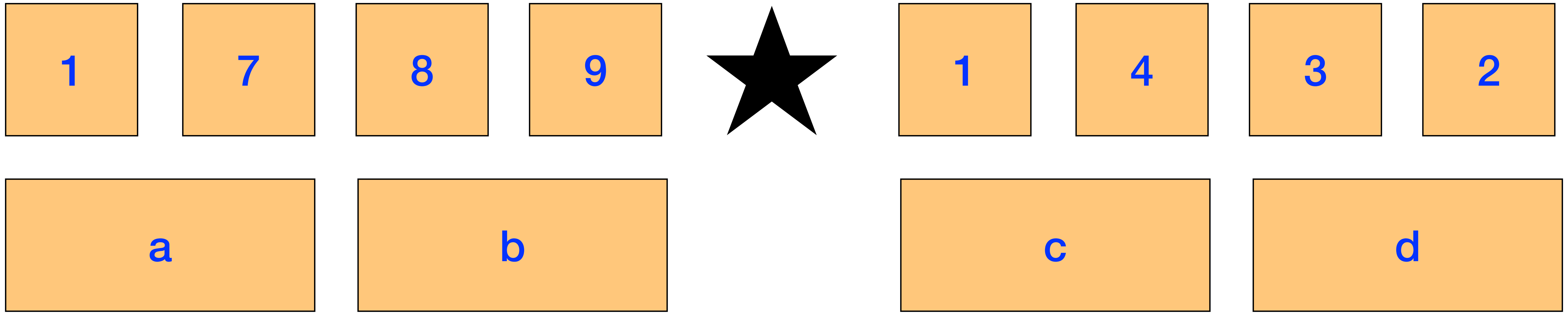

1 1
1 7 8 9
☆ ☆ ☆ ☆
1 4 3 2
(n-1)(n+1) ✢
3 5 7 8
n ☆ n-1 ✢
5 3 6 7
n ☆ n-1 ✢
7 1 5 6
n ☆ n-1 ✢
1 7 8 9
n ☆ n-1 ✢

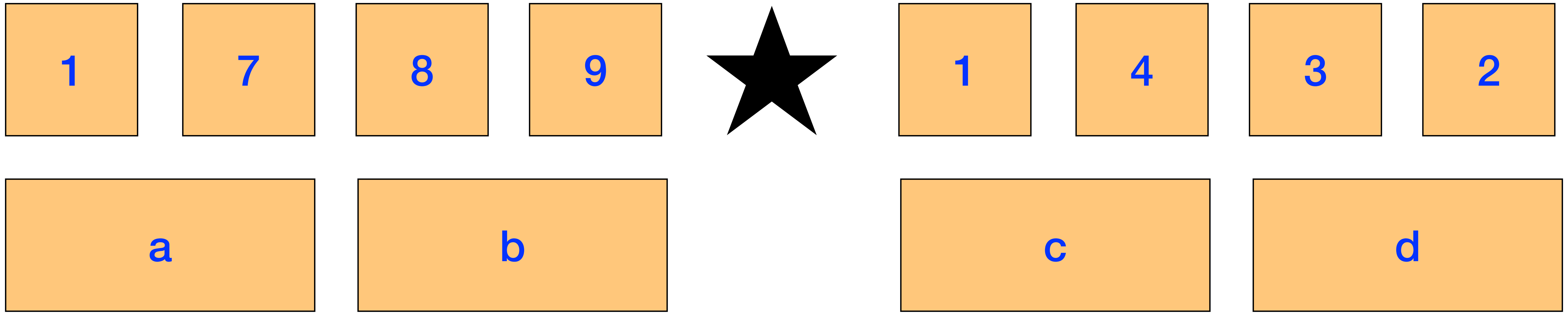
$$\Theta(n^2)$$



Theme 1


1
7
8
9
1
4
3
2

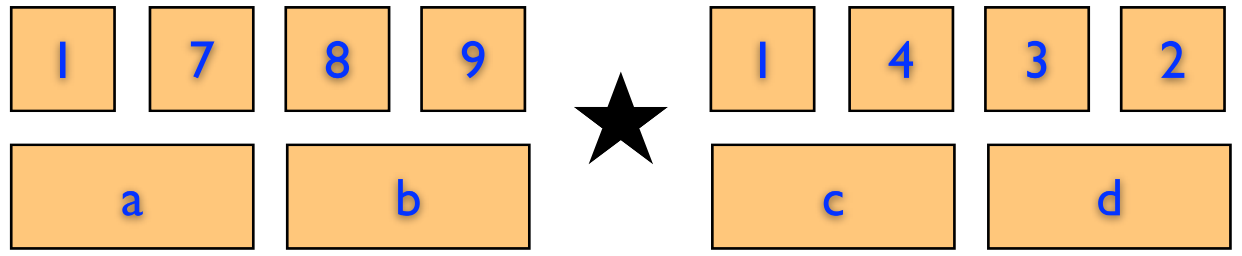

1
7
8
9
1
4
3
2
a
b
c
d



$$ac100^2 + (ad + bc)100 + bd$$

n-digit inputs

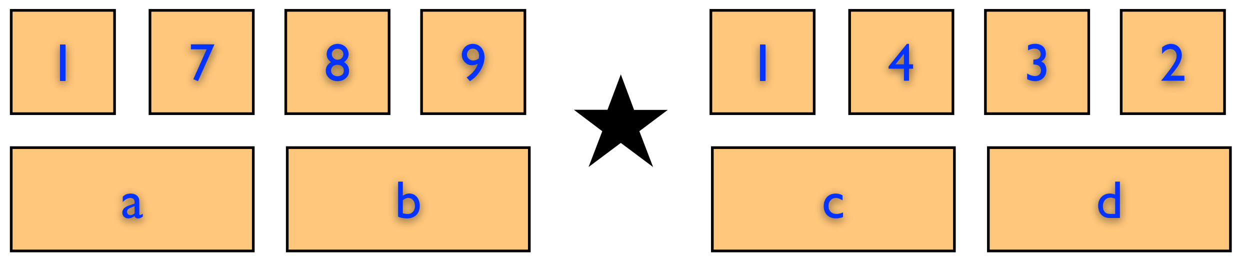
Mult(ab, cd)



$$ac100^2 + (ad + bc)100 + bd$$

Base case: return b*d if inputs are 1-digit n-digit inputs

Mult(ab, cd)



$$ac100^2 + (ad + bc)100 + bd$$

Base case: return b*d if inputs are 1-digit

Else:

Compute x = Mult(a,c)

Compute y = Mult(a,d)

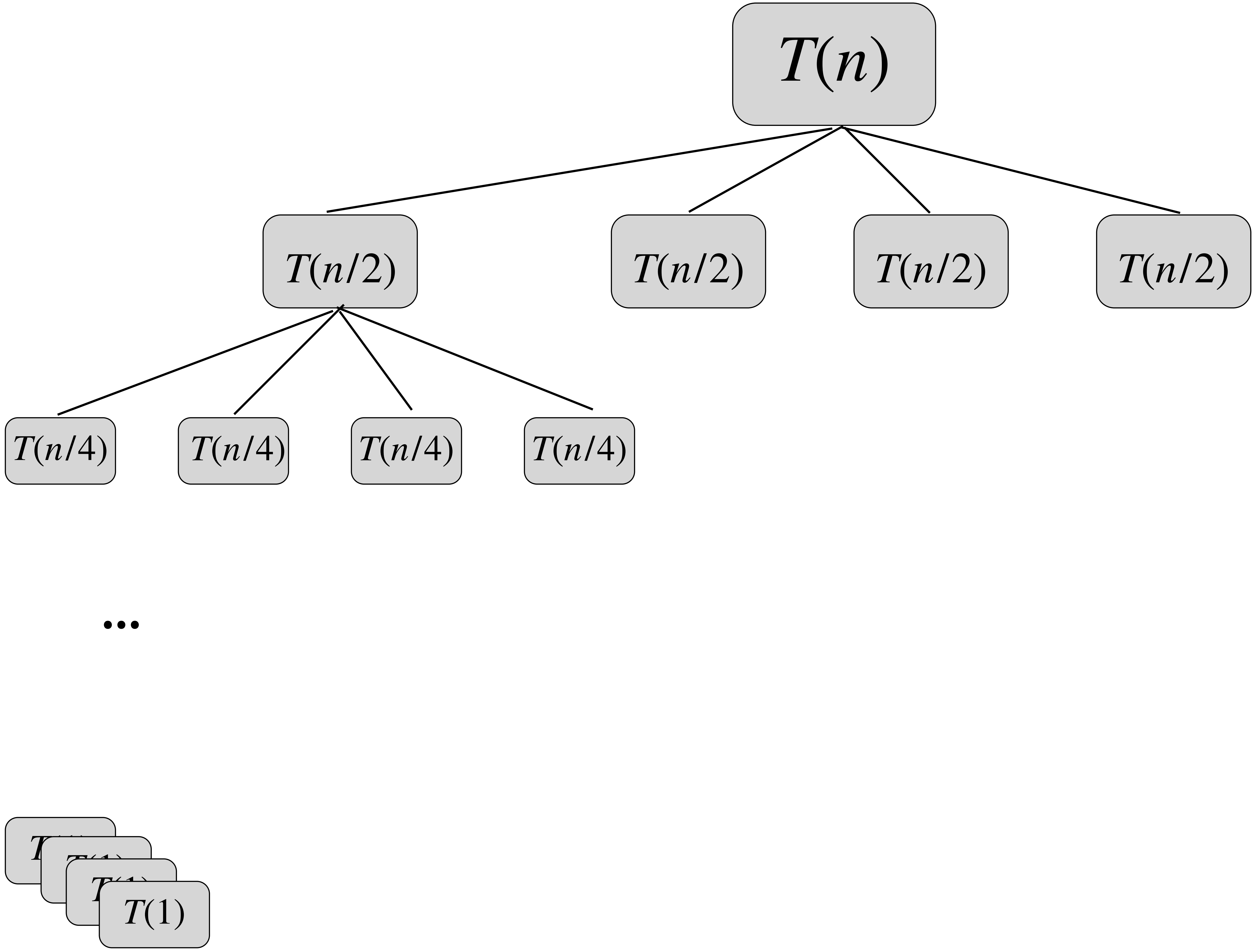
Compute z = Mult(b,c)

Compute w = Mult(b,d)

Return $r = x*10^n + (y+z)10^{n/2} + w$

$$T(n) = 4T(\lceil n/2 \rceil) + 3n$$

$$T(n) = 4T(\lceil n/2 \rceil) + 3n$$



$3n$

$4 \cdot (3n/2)$

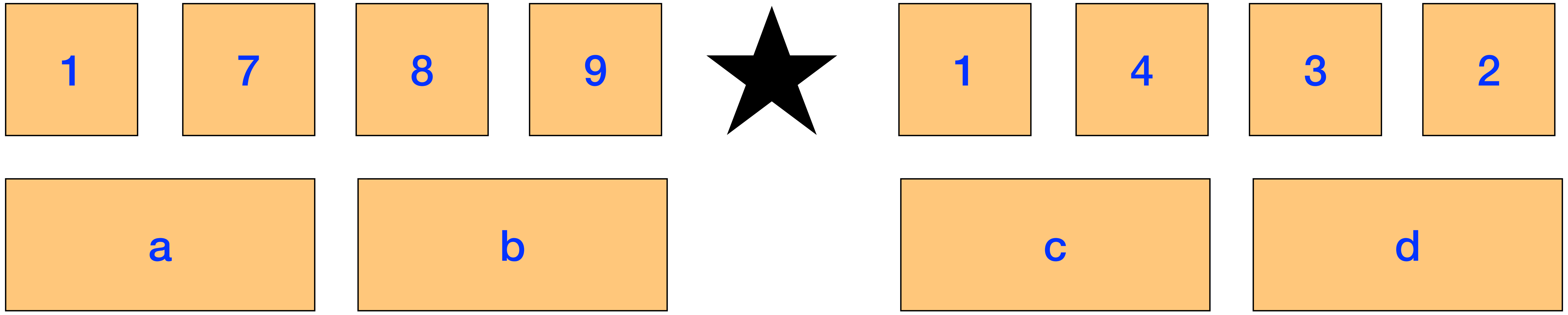
$16 \cdot (3n/4)$

$4^{\lceil \log n \rceil} \cdot (3n/2^{\lceil \log n \rceil})$

calculations:

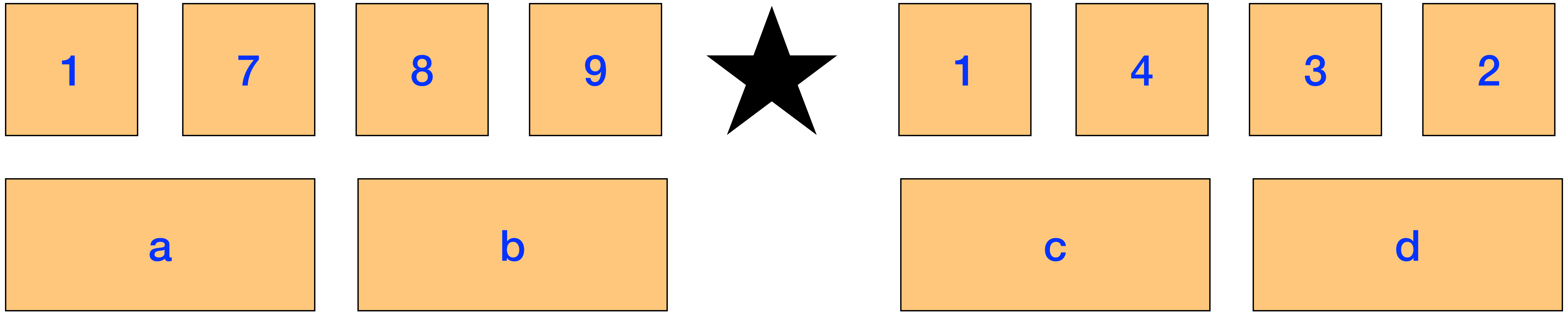
How can we improve?

Karatsuba Algorithm



$$ac100^2 + (ad + bc)100 + bd$$

Karatsuba Algorithm

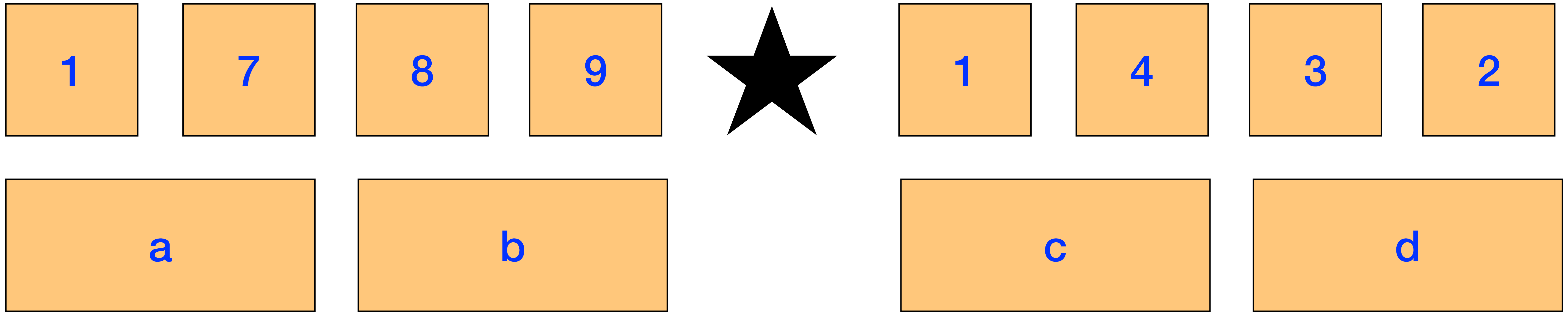


$$ac100^2 + (ad + bc)100 + bd$$

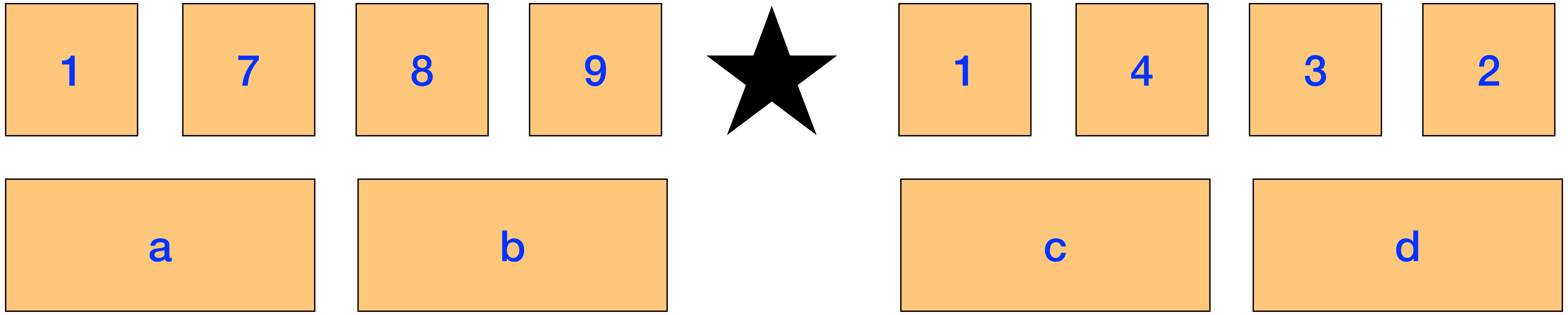
$$(a + b)(c + d) = ac + ad + bc + bd$$

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

Karatsuba Algorithm



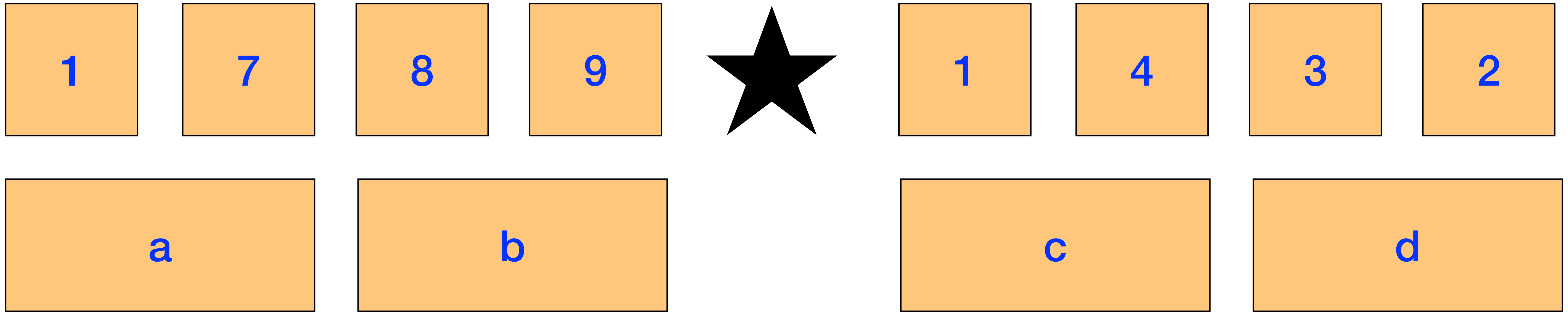
Karatsuba Algorithm



Recursively compute

1. $ac, bd, (a + b)(c + d)$

Karatsuba Algorithm

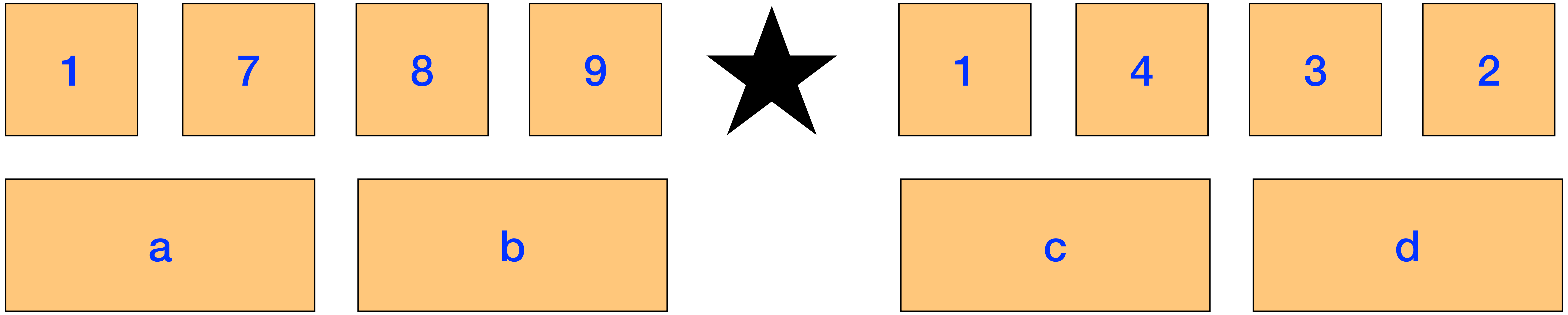


Recursively compute

❶ $ac, bd, (a+b)(c+d)$

❷ $ad + bc = (a+b)(c+d) - ac - bd$

Karatsuba Algorithm



Recursively compute

1. $ac, bd, (a+b)(c+d)$
2. $ad + bc = (a+b)(c+d) - ac - bd$
3. $ac100^2 + (ad+bc)100 + bd$

Karatsuba(ab, cd)

Base case: return b*d if inputs are 1-digit

ac = Karatsuba(a,c)

bd = Karatsuba(b,d)

t = Karatsuba((a+b),(c+d))

mid = t - ac - bd

RETURN $ac*100^2$ + mid*100 + bd

Karatsuba(ab, cd)

Base case: return b*d if inputs are 1-digit

ac = Karatsuba(a,c)

bd = Karatsuba(b,d)

t = Karatsuba((a+b),(c+d))

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

Ignoring issue of carries

mid = t - ac - bd

$4n$

RETURN ac*100^2 + mid*100 + bd

$4n$

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

calculations:

calculations:

$$T(n) = 9n + \left(\frac{3}{2}\right) \cdot 9n + \left(\frac{3}{2}\right)^2 \cdot 9n + \cdots + \left(\frac{3}{2}\right)^{\lceil \log_2 n \rceil} \cdot 9n$$

$$= 9n \left[1 + \left(\frac{3}{2}\right) + \left(\frac{3}{2}\right)^2 + \cdots + \left(\frac{3}{2}\right)^{\log_2 n} \right] = 9n \left[\frac{\left(\frac{3}{2}\right)^{\lceil \log_2 n \rceil + 1} - 1}{\left(\frac{3}{2}\right) - 1} \right]$$

$$= (9n)(2) \left[\left(\frac{3}{2}\right)^{\log_2 n + 1} - 1 \right]$$

$$3 = 2^{\log_2 3}$$

$$= (9n)(2)\left(\frac{3}{2}\right) \left[\frac{3^{\log_2 n}}{2^{\log_2 n}} \right] - 18n$$

n

$$= 27 \cdot \frac{3^{\log_2 n}}{\text{la } \kappa \text{a}} - 18n = 27 \cdot n^{\log_2 3} - 18n = O(n^{\log_2 3})$$

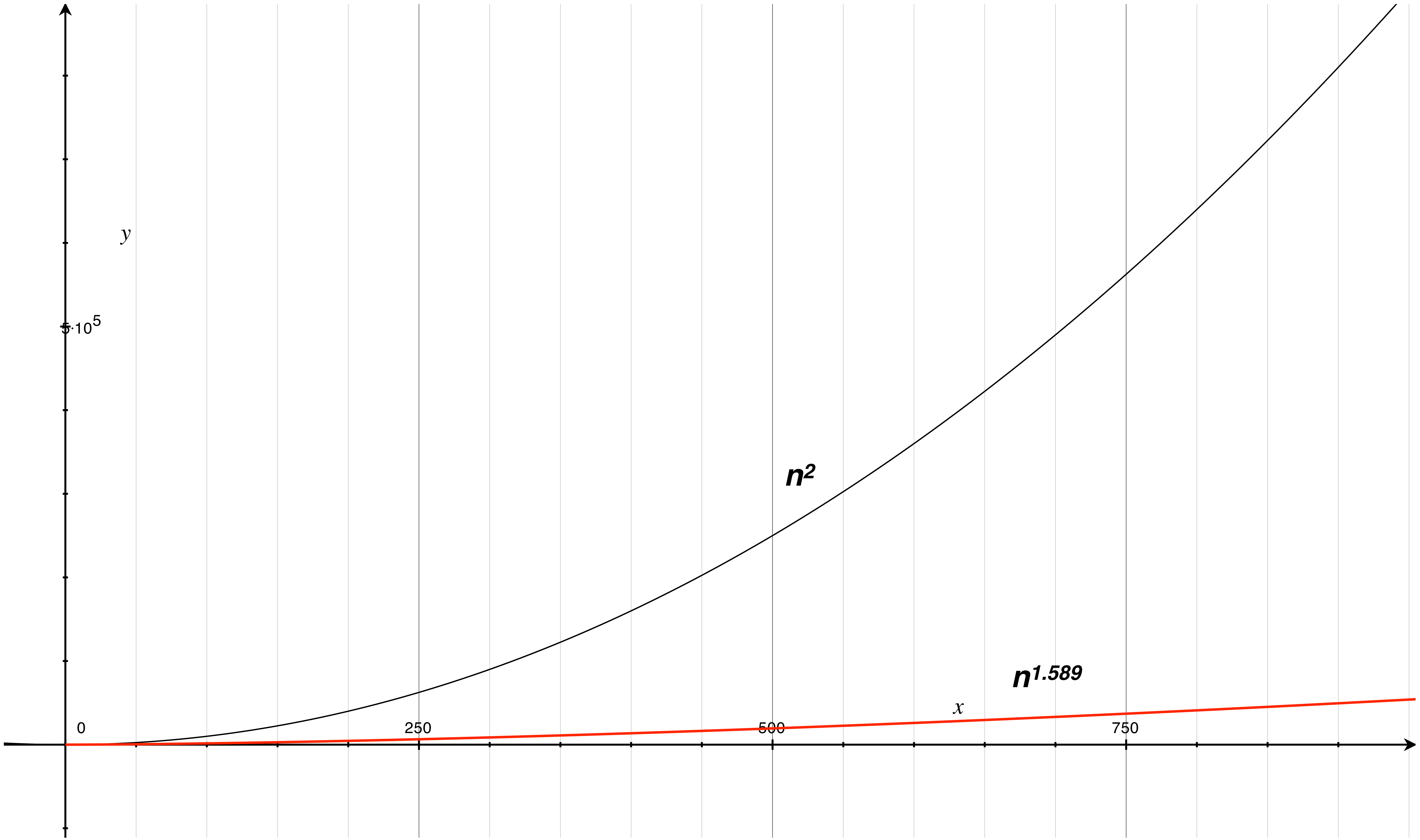
$$= \left(2^{\log_2 3}\right)^{\log_2 n} = \left(2^{\log_2 n}\right)^{\log_2 3} = \left(n^{\log_2 3}\right)$$

$$T(n) = 3T(n/2) + 9n$$

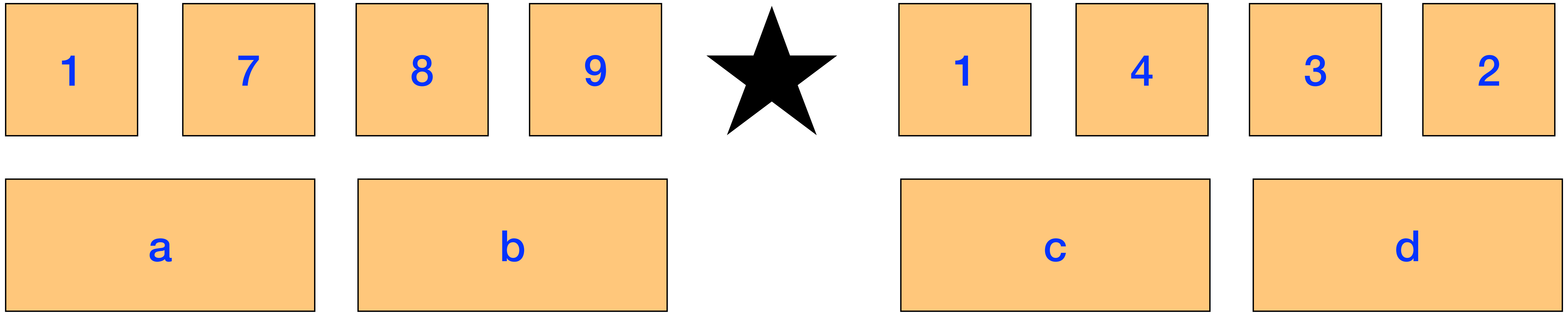
$$O(n^{\log_2(3)})$$

$$T(n) = 3T(n/2) + 9n$$

$$O(n^{\log_2(3)}) \quad O(n^{1.589})$$



y
5·10^5
n^2
n^1.589
x
0
250
500
750



$$T(n) = 3T(n/2) + 9n$$

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

simpler proof technique?

❶ induction redux

classic goal: prove that some property P(k) is true for all k

$\forall k, P(k)$ holds

❶ one long proof...

classic goal: prove that some property P(k) is true for all k

$\forall k, P(k)$ holds

❶ Induction

classic
base case: $P(1)$ is true.

classic
inductive step:

$$\left.\begin{matrix} P(1) \\ \cdots \\ P(k) \end{matrix}\right\} \text{ implies } P(k+1) \text{ is true}$$

❶ Induction, asymptotic style

classic
base case: $P(n^*)$ is true.

classic
inductive step: $\left.\begin{matrix} P(n^*) \\ \cdots \\ P(k) \end{matrix}\right\}$ implies $P(k+1)$ is true

simpler proof **(guess +chk)**

$$T(n) = 3T(n/2) + 9n$$

simpler proof

simpler proof

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

Induction hypothesis: `T(n) < dn`$^{1.59}$

It is true for n=1. suppose it is true for n<n0.

$$T(n_0+1) = 3T((n_0+1)/2) + c(n_0+1)$$

$$< 3d[(n_0+1)/2]^{1.59} + c(n_0+1)$$ By the induction hypothesis

$$< 3/2^{1.59} d(n_0+1)^{1.59} + c(n_0+1)$$

$$< 0.997 d(n_0+1)^{1.59} + c(n_0+1)$$

Another example: sorting

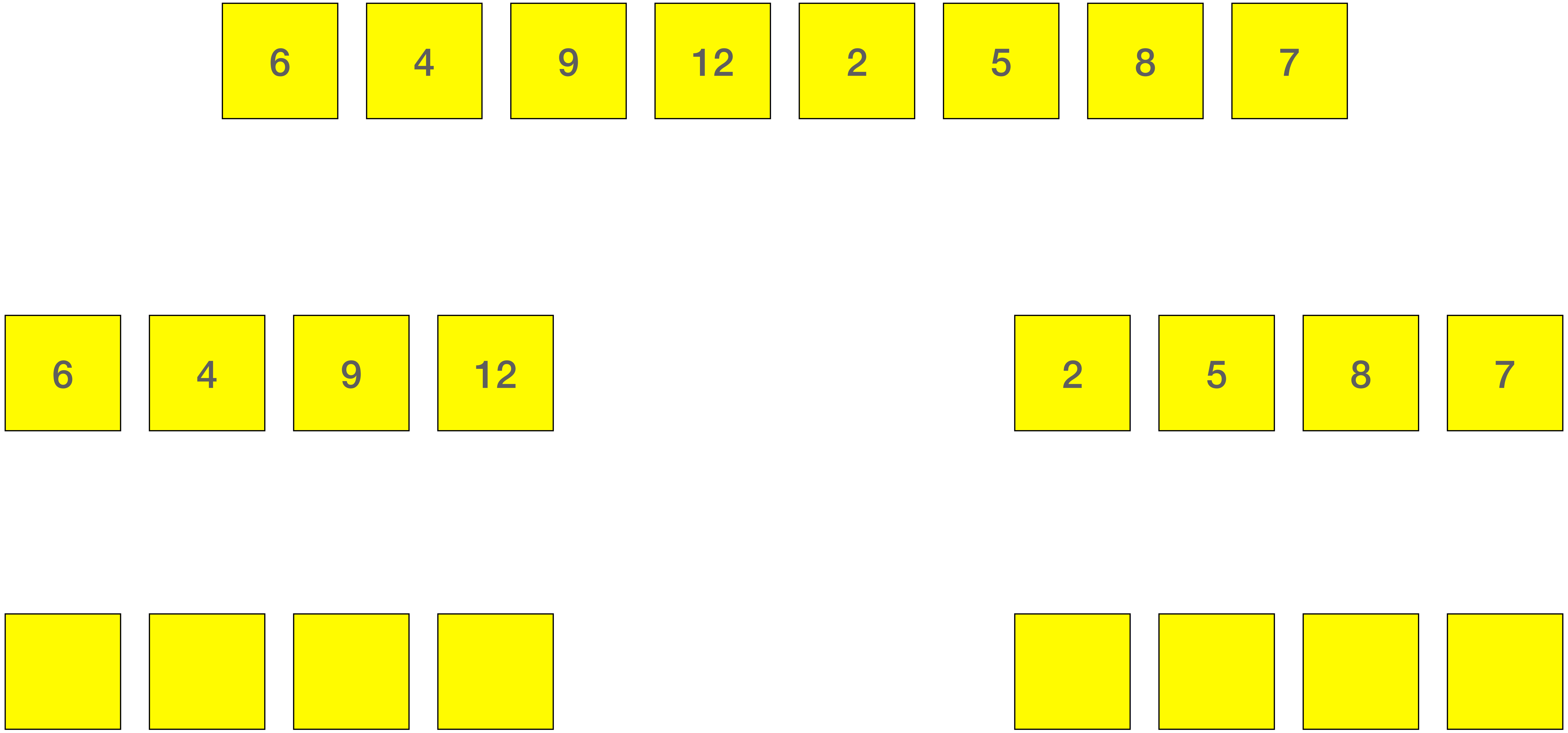
mergesort

goal:

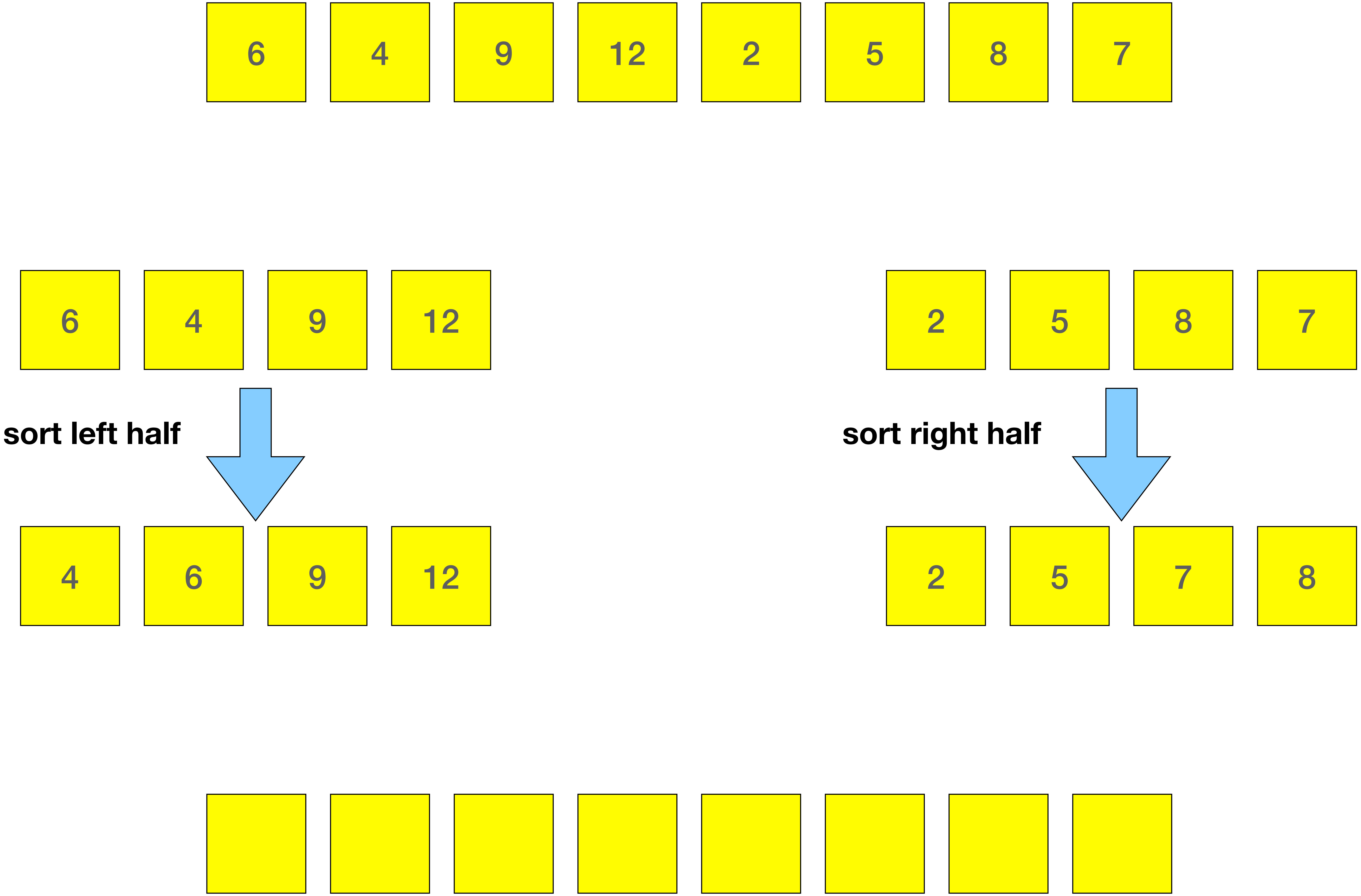
technique:

6	4	9	12	2	5	8	7

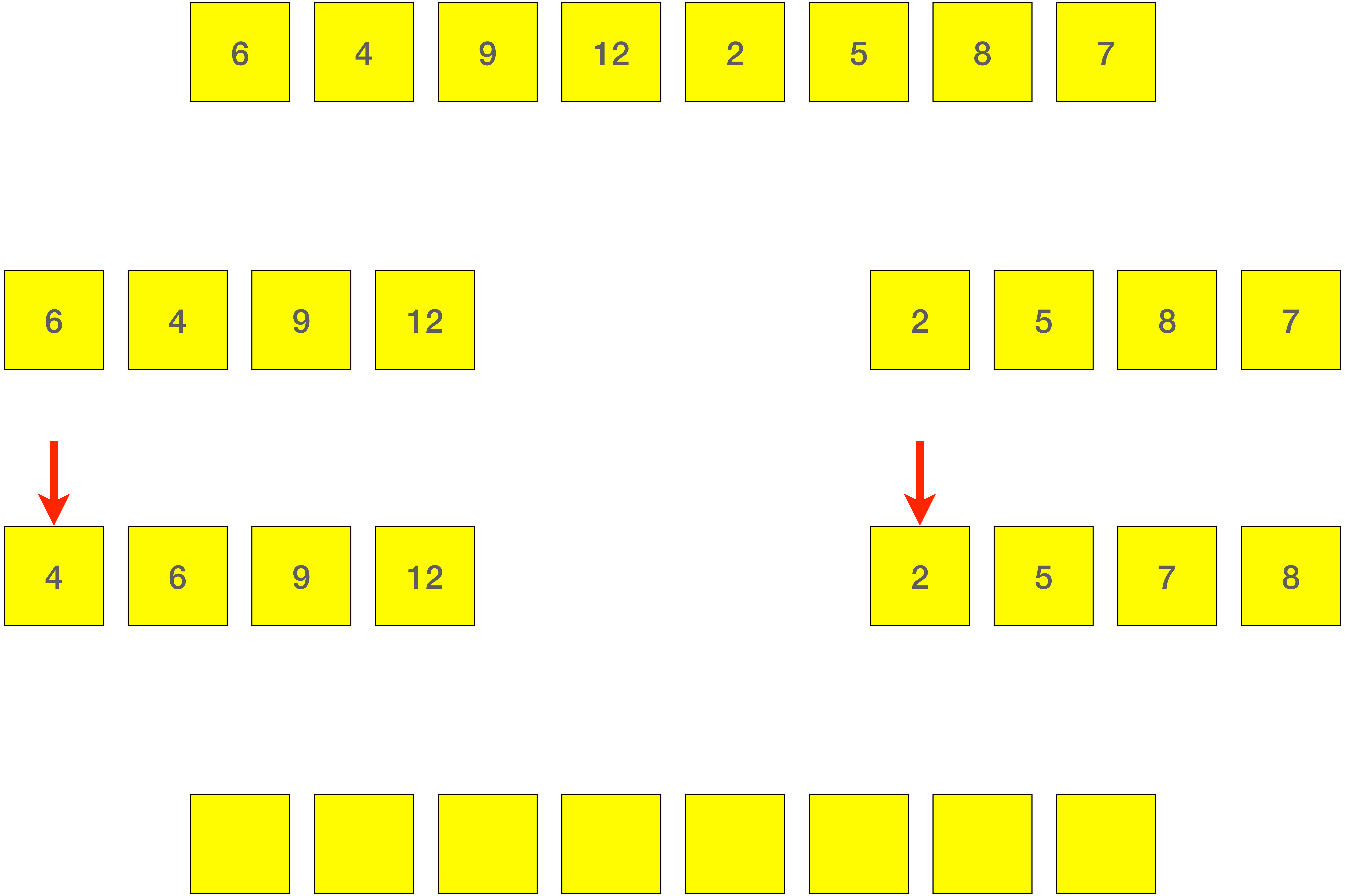
mergesort



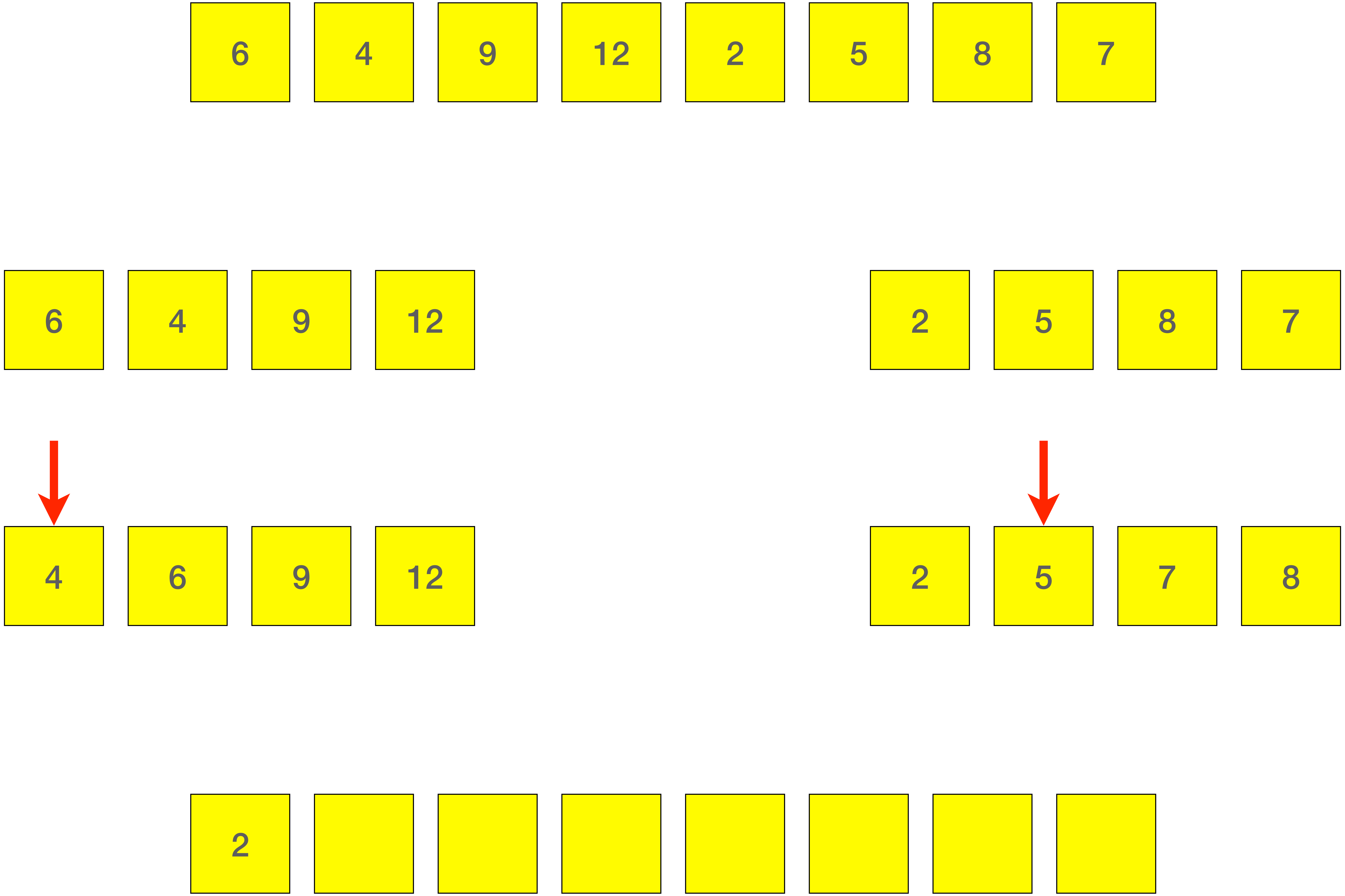
mergesort


6
4
9
12
2
5
8
7
6
4
9
12
2
5
8
7
sort left half
sort right half
4
6
9
12
2
5
7
8

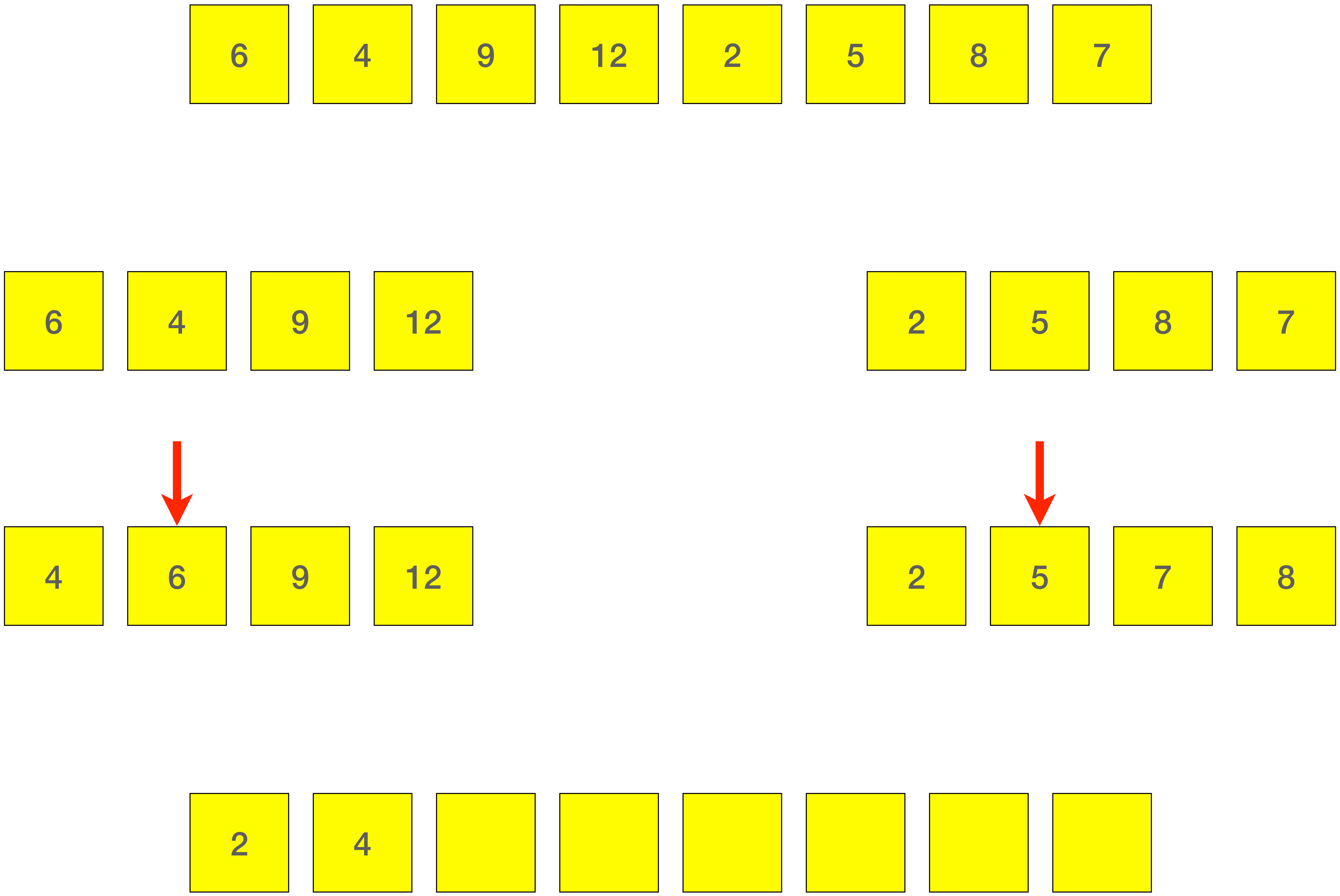
mergesort



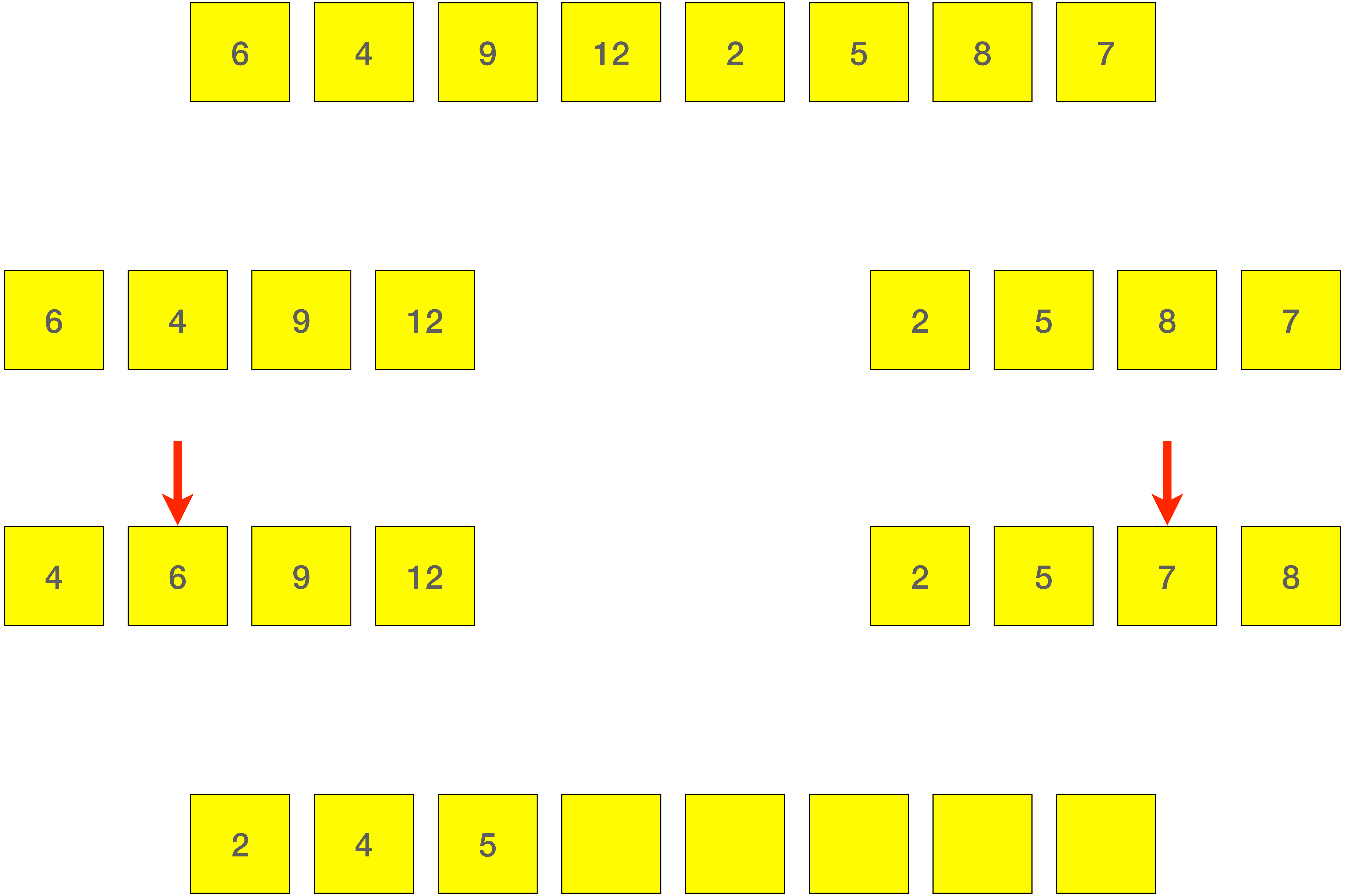
mergesort



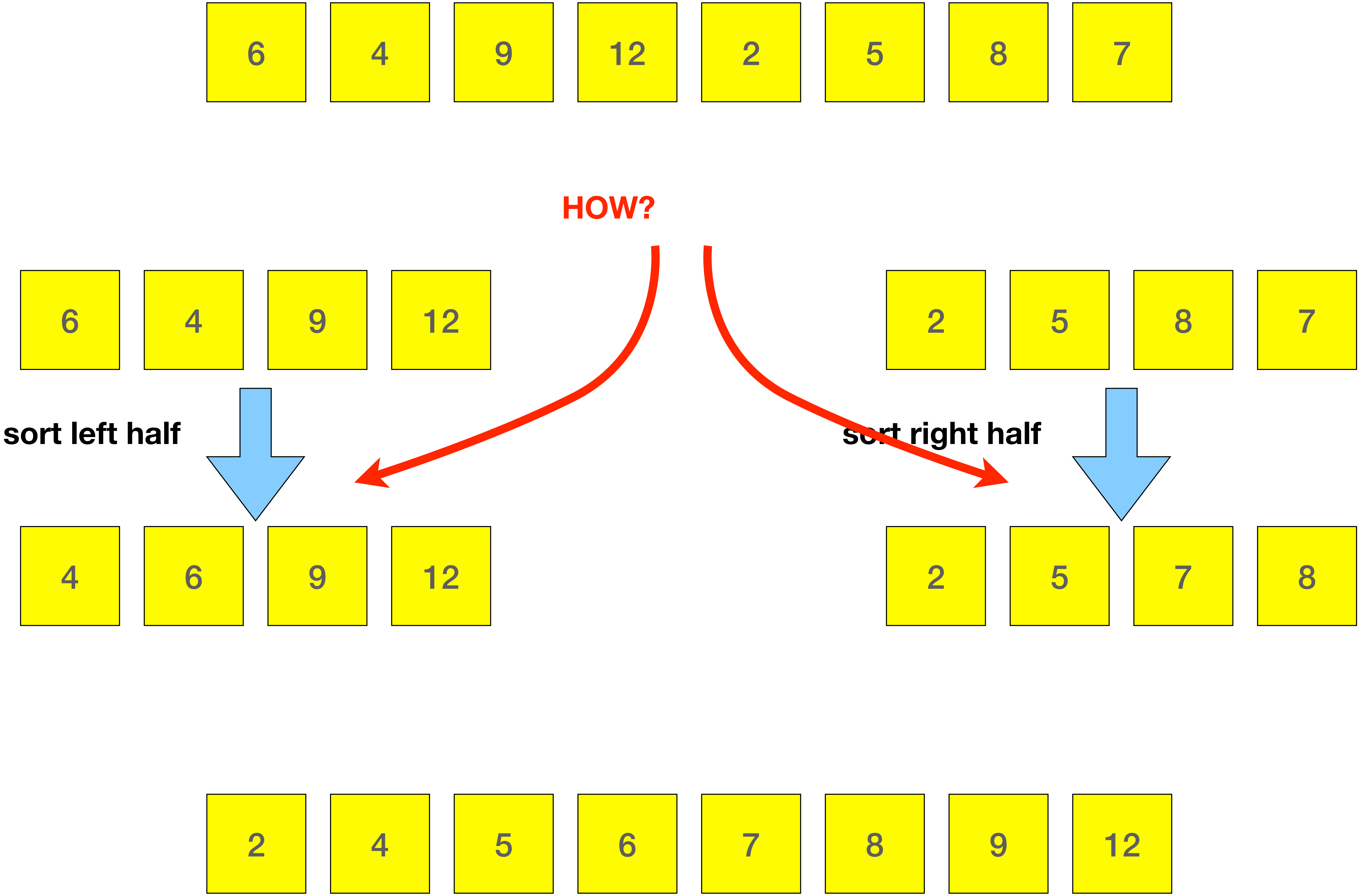
mergesort



mergesort



mergesort


6 4 9 12 2 5 8 7
HOW?
6 4 9 12
2 5 8 7
sort left half
sort right half
4 6 9 12
2 5 7 8
2 4 5 6 7 8 9 12

mergesort(`A,start,end`)

❶

❷

❸

❹

❺

mergesort(`A,start,end`)

1. **`if start < end`**
2. $q \leftarrow \lfloor (\text{start} + \text{end})/2 \rfloor$
3. `mergesort(A,start,q)`
   `mergesort(A,q+1,end)`
4. `merge(A,start,q,end)`
5. `else ...`

mergesort(A,start,end)

1. **if start < end**
2. $q \leftarrow \lfloor(\text{start} + \text{end})/2\rfloor$
3. `mergesort(A,start,q)`
   `mergesort(A,q+1,end)`
4. `merge(A,start,q,end)`
5. `else ...`

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

mergesort(`A,start,end`)

Running time?

1. `if start < end`
2. $q \leftarrow \lfloor (\text{start} + \text{end})/2 \rfloor$
3. `mergesort(A,start,q)`
   `mergesort(A,q+1,end)`
4. `merge(A,start,q,end)`
5. `else ...`

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

show:

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

prove:

hypothesis:

base case:

inductive step:

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

prove: $T(n) = O(n \log n)$

property: $T(n) < cn \log n$ **for c>1**

base case:

inductive step:

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

goal is to show $T(n) = \Theta(n \log n)$

show: $T(n) \le n \log n$

Proof: Base case holds for $n \le 5$. Assume that the hypothesis holds for all $k \le n$. Consider

$$T(n+1) = 2T\left(\frac{n+1}{2}\right) + (n+1)$$

$$\le 2\left(\frac{n+1}{2}\right)\log\left(\frac{n+1}{2}\right) + n+1$$

$$= (n+1)\left[\log(n+1) - 1\right] + n+1$$

$$= (n+1)\log(n+1) - (n+1) + n+1$$

$$= (n+1)\log(n+1)$$

$\frac{n+1}{2} < n$, $\Rightarrow T\left(\frac{n+1}{2}\right) \le \left(\frac{n+1}{2}\right)\log\left(\frac{n+1}{2}\right)$ by ind hypothesis

$$\log\left(\frac{a}{b}\right) = \log(a) - \log(b)$$