L3 4102

sep 2 2013

shelat

Karatsuba algorithm

n digit number

1 7 8 9 ★ 1 4 3 2

n/2 digit number

a b c d

We are aiming to compute
ac*100^2 + (ad+bc)*100 + bd.

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

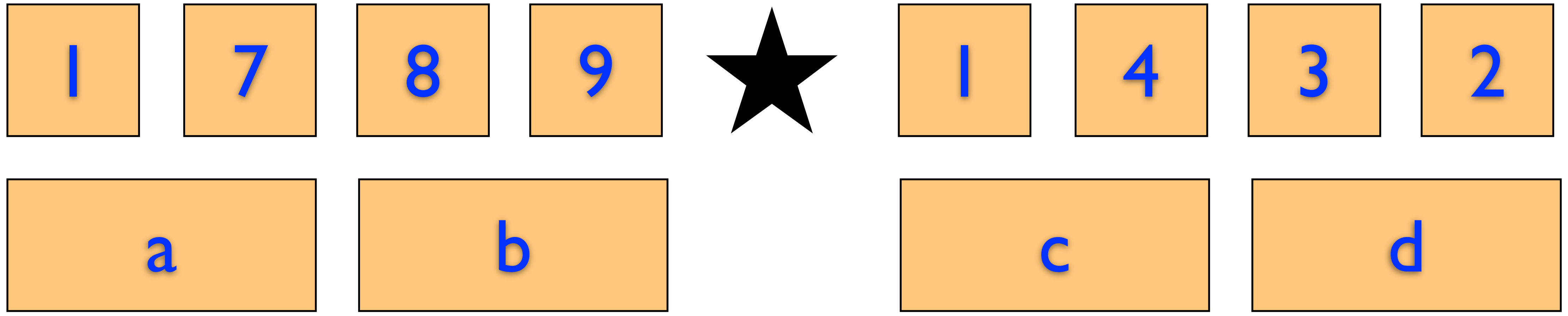
① Recursively compute $a \cdot c$ $b \cdot d$ $(a+b)(c+d)$ $3T(\frac{n}{2})$ $2\,O(n)$ $2A(n)$

② $(ad+bc) = (a+b)(c+d) - a \cdot c - b \cdot d$ $2A(2n) = 4O(n)$

The way that we are going to compute
(ad+bc) is to (1) compute (a+b)(c+d)
recursively, (2) then we subtract of ac
bd

③ $ac \cdot (100)^2 + (ad+bc) \cdot 100 + bd$ $2A(n) = 2O(n)$

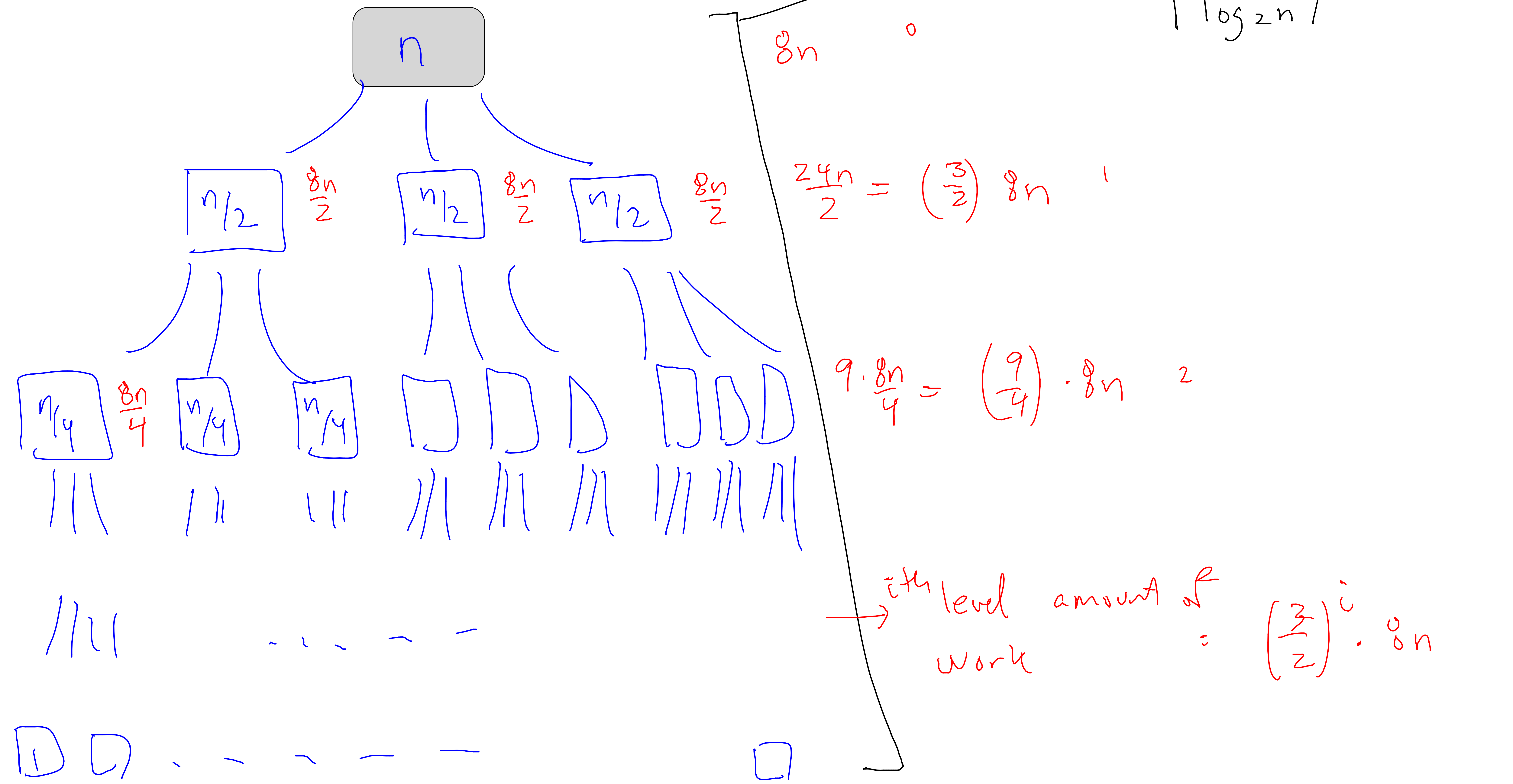
Karatsuba algorithm



Recursively compute

1. $ac, bd, (a+b)(c+d)$ $\quad 3T(n/2) + 2O(n)$
2. $ad + bc = (a+b)(c+d) - ac - bd$ $\quad 2O(n)$ $\approx 2n$
3. $ac100^2 + (ad+bc)100 + bd$ $\quad 2O(n)$

$$T(n) = 3T(n/2) + 8O(n)$$


how many levels of recursion?
$\lceil \log_2 n \rceil$
n
n/2
n/2
n/2
8n/2
8n/2
8n/2
n/4
n/4
n/4
8n/4
8n
0
24n/2 = (3/2) 8n
1
9 · 8n/4 = (9/4) · 8n
2
ith level amount of work = (3/2)^i · 8n
1

calculations:

Total work: $8n + \left(\frac{3}{2}\right) 8n + \left(\frac{3}{2}\right)^2 8n + \cdots + \left(\frac{3}{2}\right)^{\lceil \log_2 n \rceil} \cdot 8n$

$$= 8n \cdot \sum_{i=0}^{\log_2 n} \left(\frac{3}{2}\right)^i = 8n \left[\frac{\left(\frac{3}{2}\right)^{\lceil \log_2 n \rceil + 1} - 1}{\frac{3}{2} - 1} \right]$$

$$= \left(\frac{3}{2}\right) 16n \left(\frac{3}{2}\right)^{\lceil \log_2 n \rceil} - 16n$$

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

$$= 24\, n \cdot n^{\log_2 3/2} - 16n$$

$$= 24\, n^{\log_2 3} - 16n = O\left(n^{\log_2 3}\right)$$

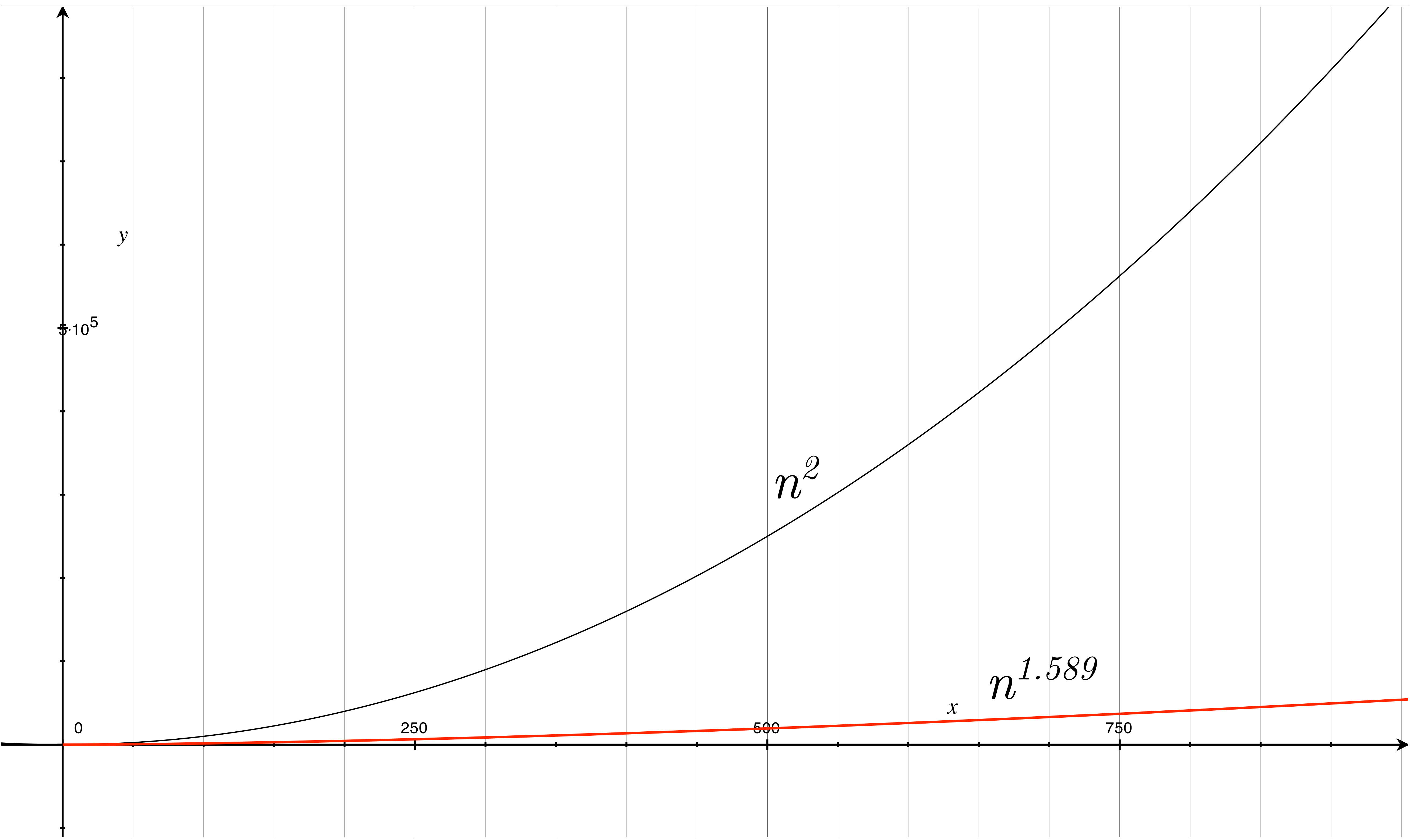
calculations:

$$T(n) = 3T(n/2) + 8O(n)$$

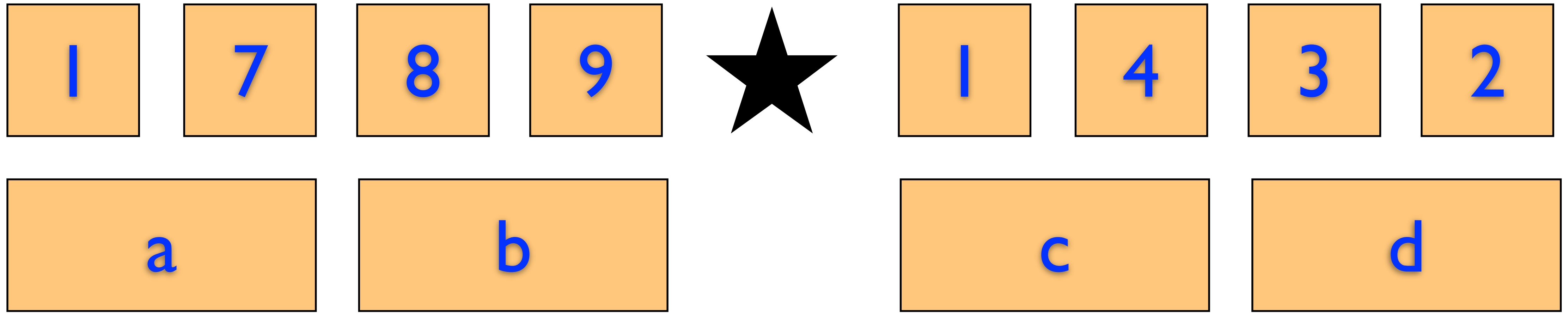
$$O(n^{\log_2(3)})$$

$$T(n) = 3T(n/2) + 8O(n)$$

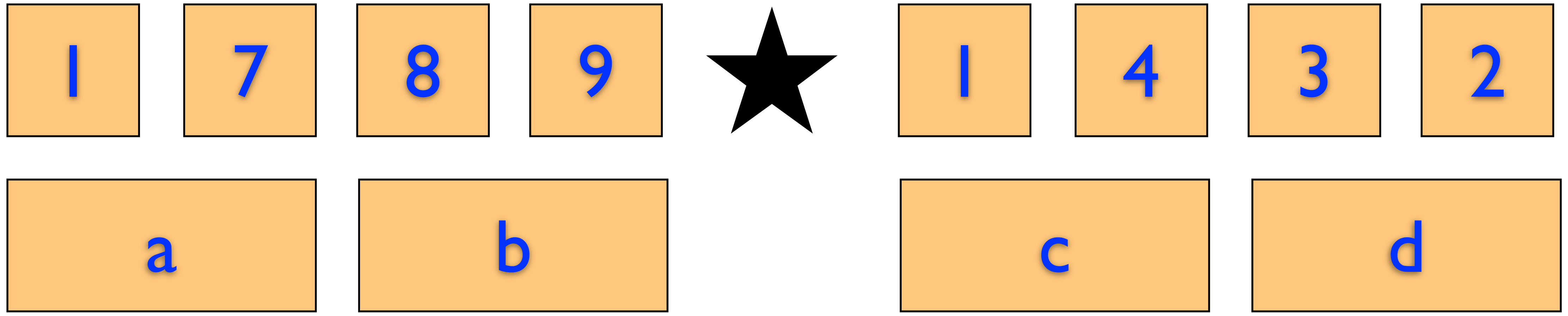
$$O(n^{\log_2(3)}) \quad O(n^{1.589})$$



y
5·10^5
0
250
500
750
x
n^2
n^1.589



$$T(n) = 3T(n/2) + 8O(n)$$



$$T(n) = 3T(n/2) + 8O(n)$$

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

simpler proof technique?

1

induction redux

classic

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

$\forall k, P(k)$ holds

❶ classic one long proof...

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

$\forall k, P(k)$ holds

❶ induction redux

classic base case: $P(1)$

classic inductive step: $P(1)$ $\cdots$ $P(k)$ true implies $P(k+1)$ true

❷ # induction redux asymptotic style

base case: $P(n^*)$

inductive step: $P(n^*)$ $\dots$ $P(k)$ true implies $P(k+1)$ true

simpler proof (guess +chk)

$$T(n) = 3T(n/2) + 8O(n)$$

Need to prove that $T(n) = O(n^{1.6})$

Need to show $T(n) < d \cdot n^{1.6}$

OK. I will show that $T(n) < 800 \cdot n^{1.6}$.

This holds for small case like 4, 5, 6.. (follows by simple inspection).

Spse that $T(n) \le 800 n^{1.6}$ for all values $\le n_0$.

Consider $T(n_0+1)$. (I need to show that $T(n_0+1) < 800(n_0+1)^{1.6}$.

Well $T(n_0+1) = 3T\left(\frac{n_0+1}{2}\right) + 8(n_0+1)$

$< 3\left[800\left(\frac{n_0+1}{2}\right)^{1.6}\right] + 8(n_0+1)$ because $\frac{n_0+1}{2} \le n_0$ & our hypothesis is applicable

$= \left(\frac{3}{2^{1.6}}\right) \cdot 800(n_0+1)^{1.6} + 8(n_0+1)$

simpler proof

$$\leq \left(\frac{3}{2^{1.6}}\right) \cdot 800(n_0+1)^{1.6} + 8(n_0+1)$$

$$< \underbrace{0.99 \cdot 800(n_0+1)^{1.6} + 0.01 \cdot 800(n_0+1)^{1.6}} + \underbrace{8(n_0+1) - 0.01 \cdot 800(n_0+1)^{1.6}}$$

$$\leq 800(n_0+1)^{1.6} \qquad \underbrace{8(n_0+1) - 8(n_0+1)^{1.6}}$$

this is negative!!

$$\Rightarrow T(n_0+1) \leq 800(n_0+1)^{1.6}.$$

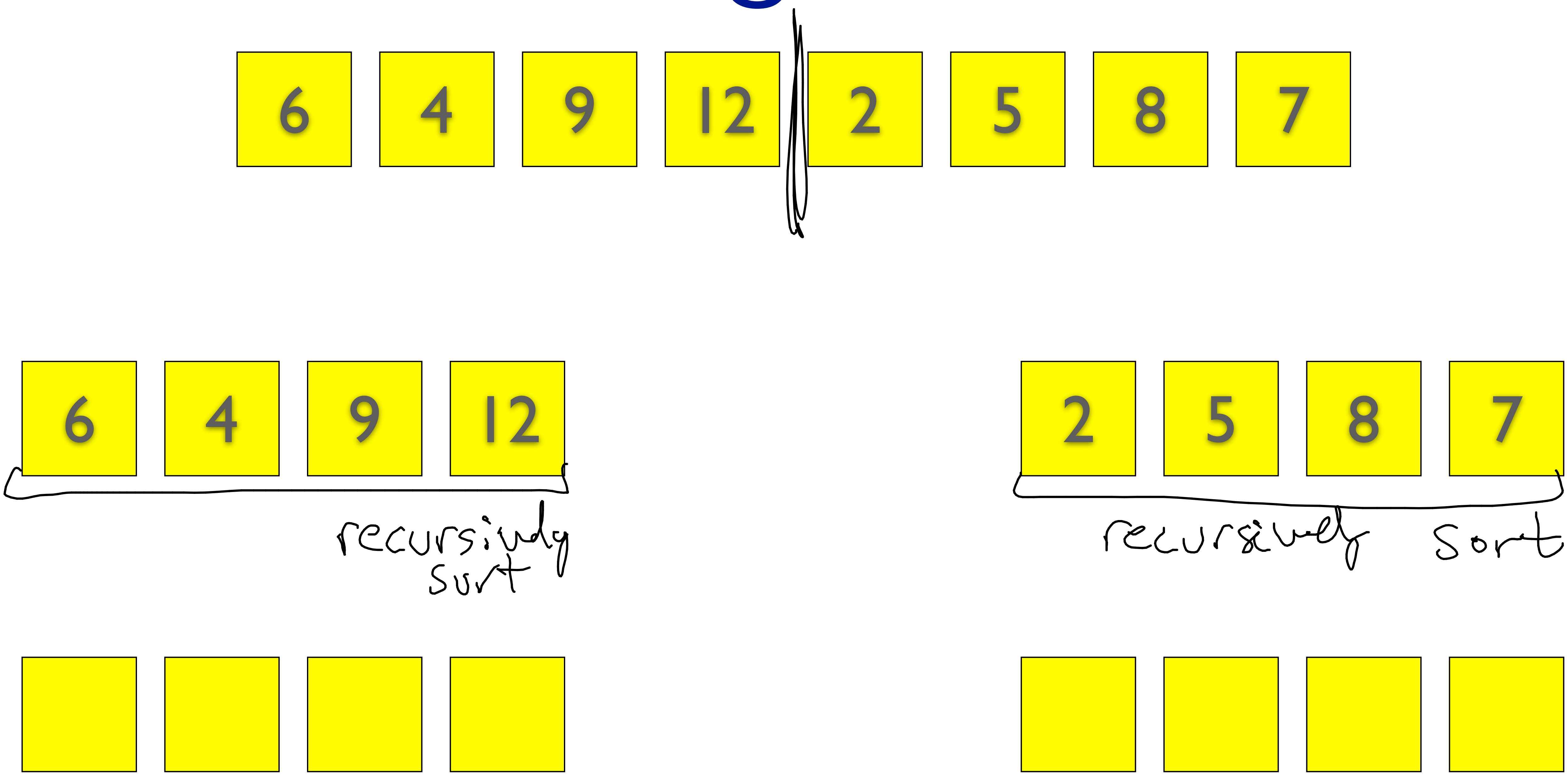
mergesort

goal: sort n-element array

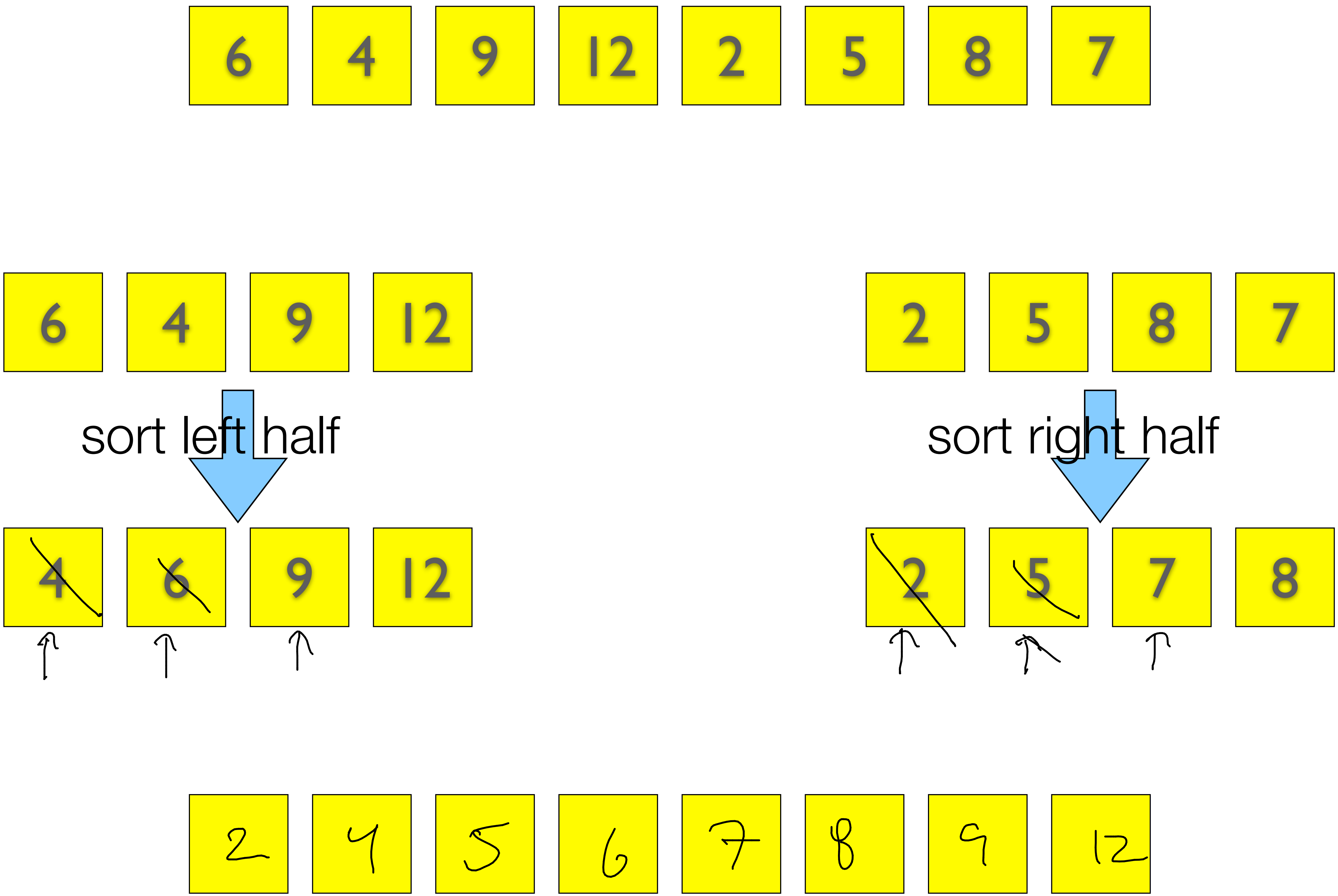
technique: ① split array into 2 halves
② sort each half ③ merge the results

6	4	9	12	2	5	8	7

mergesort

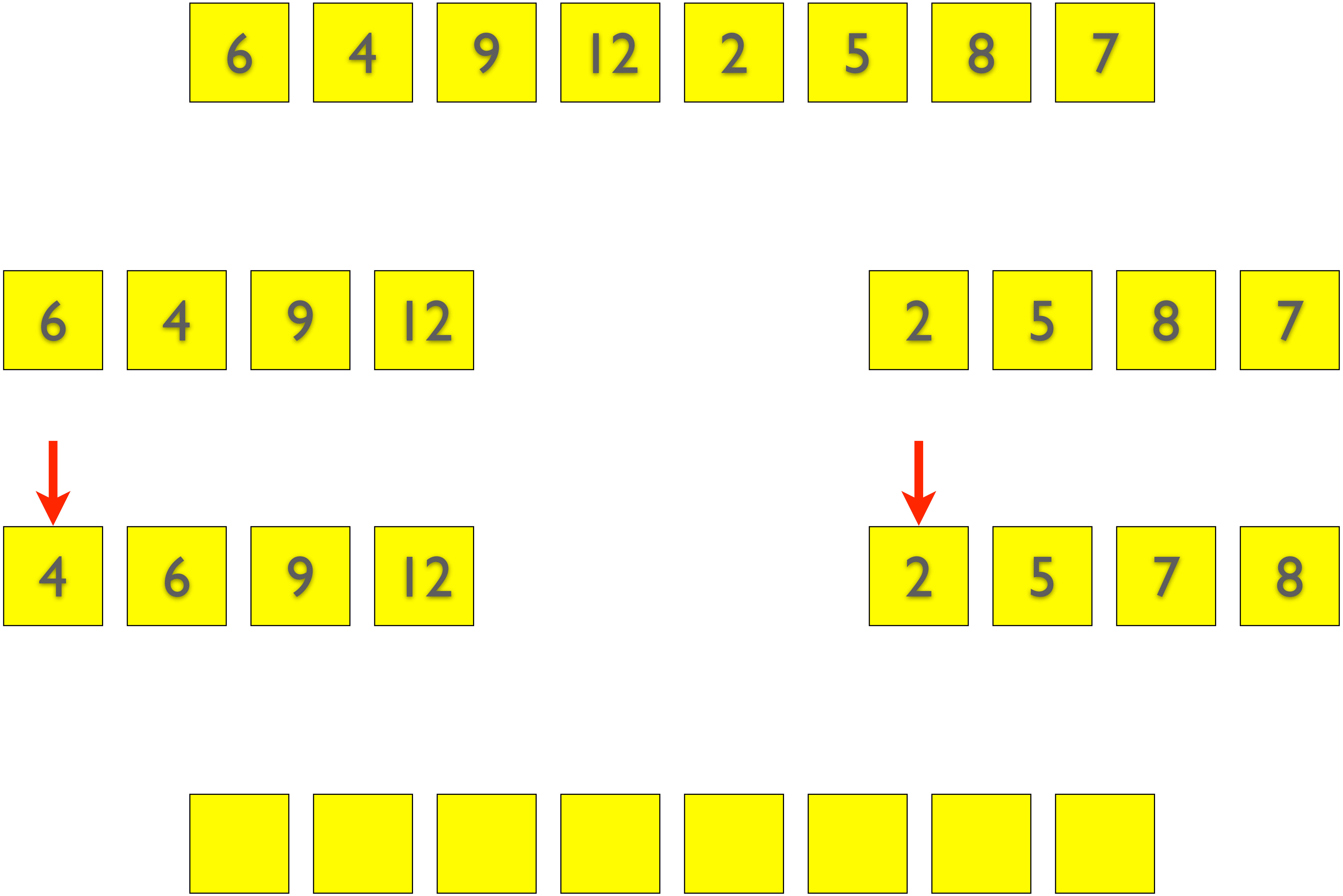


mergesort

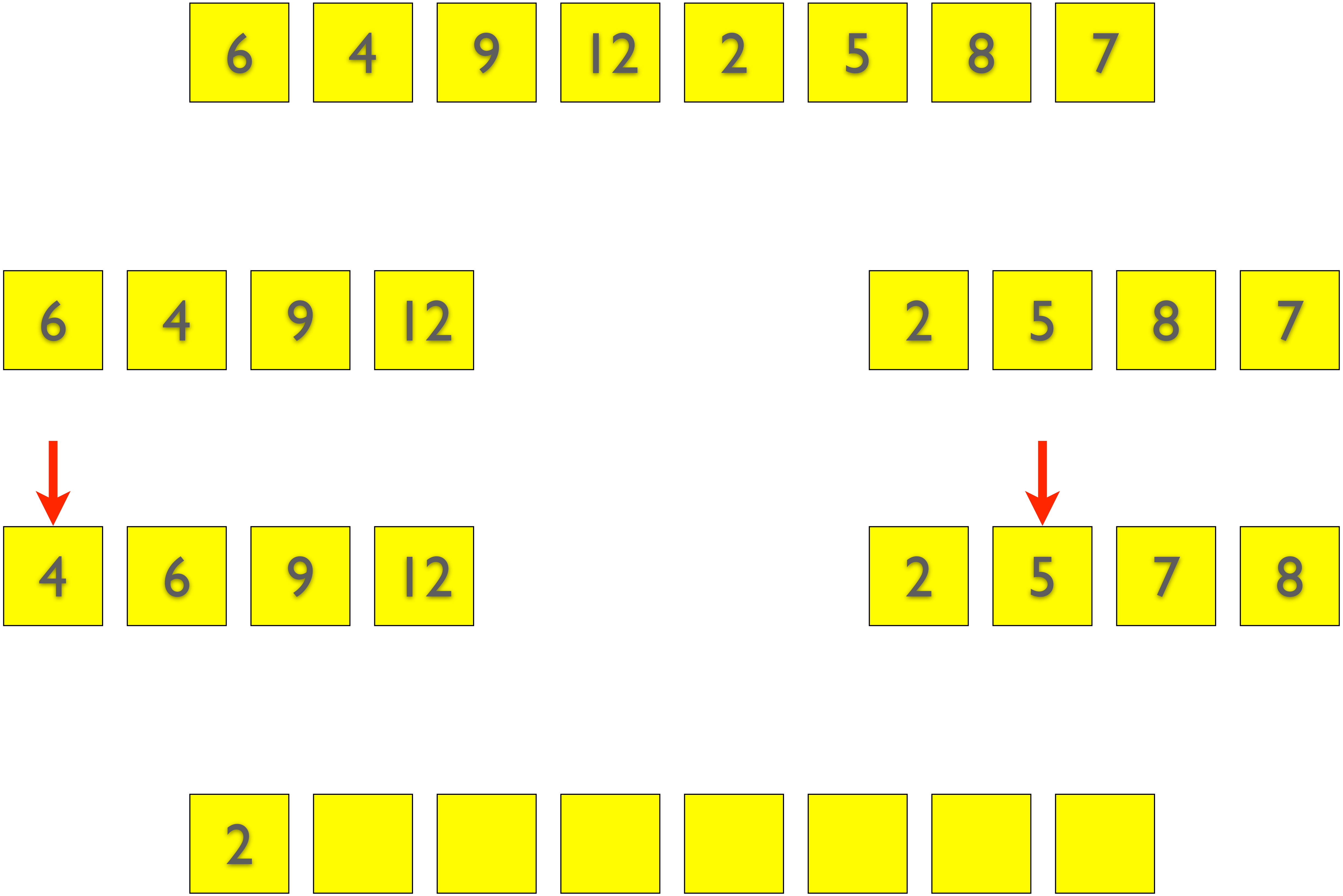


6 4 9 12 2 5 8 7
6 4 9 12
2 5 8 7
sort left half
sort right half
4 6 9 12
2 5 7 8
2 4 5 6 7 8 9 12

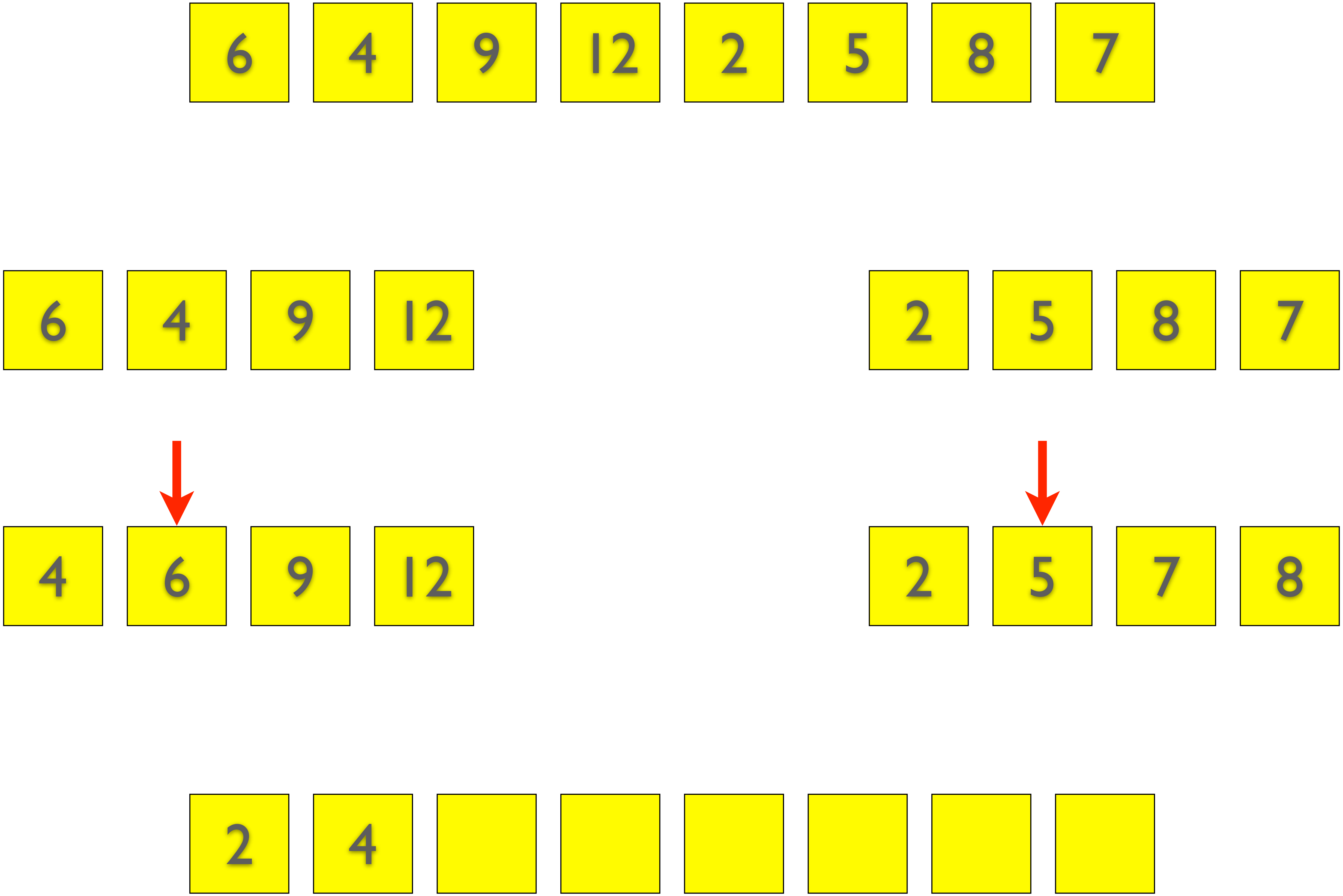
mergesort



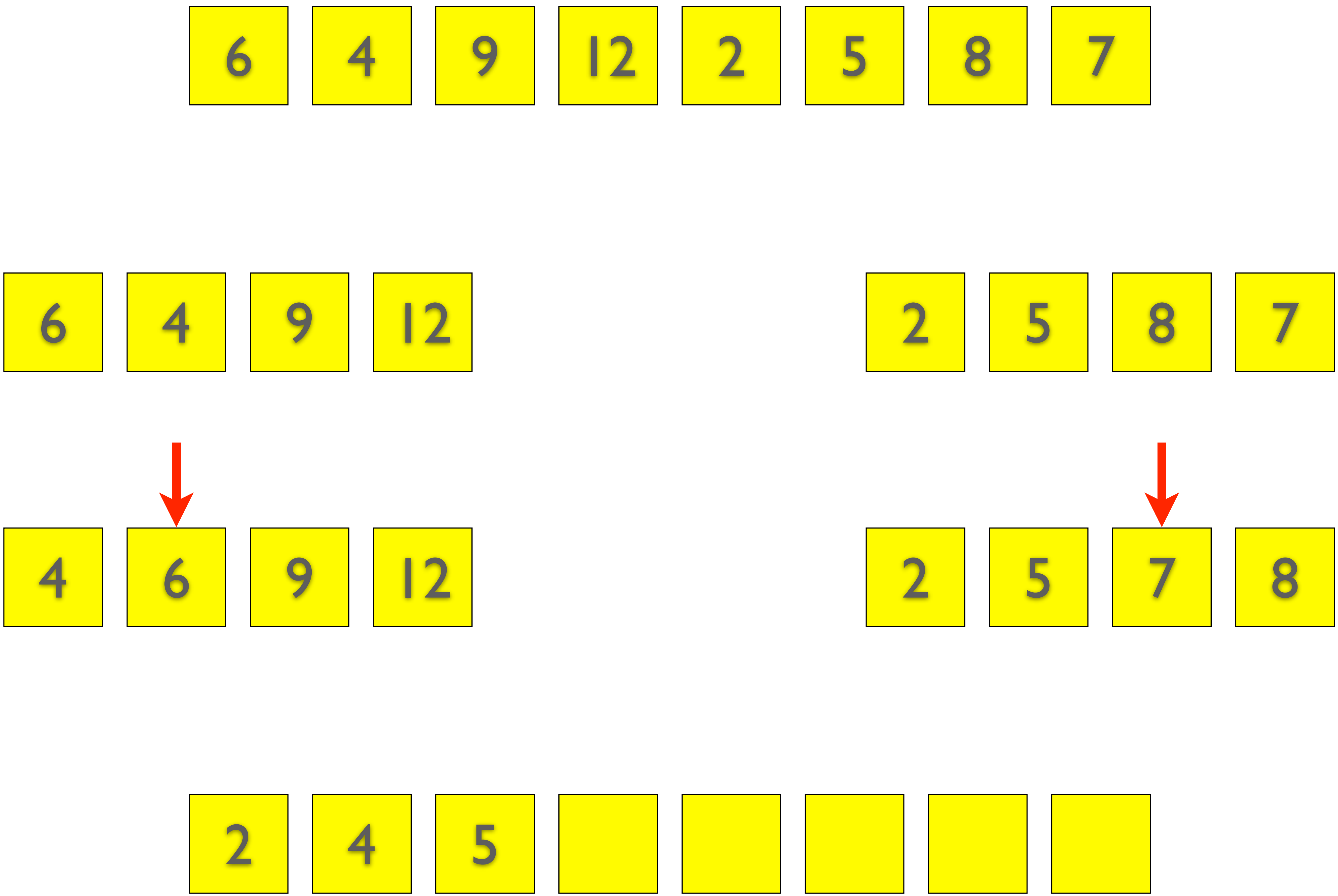
mergesort



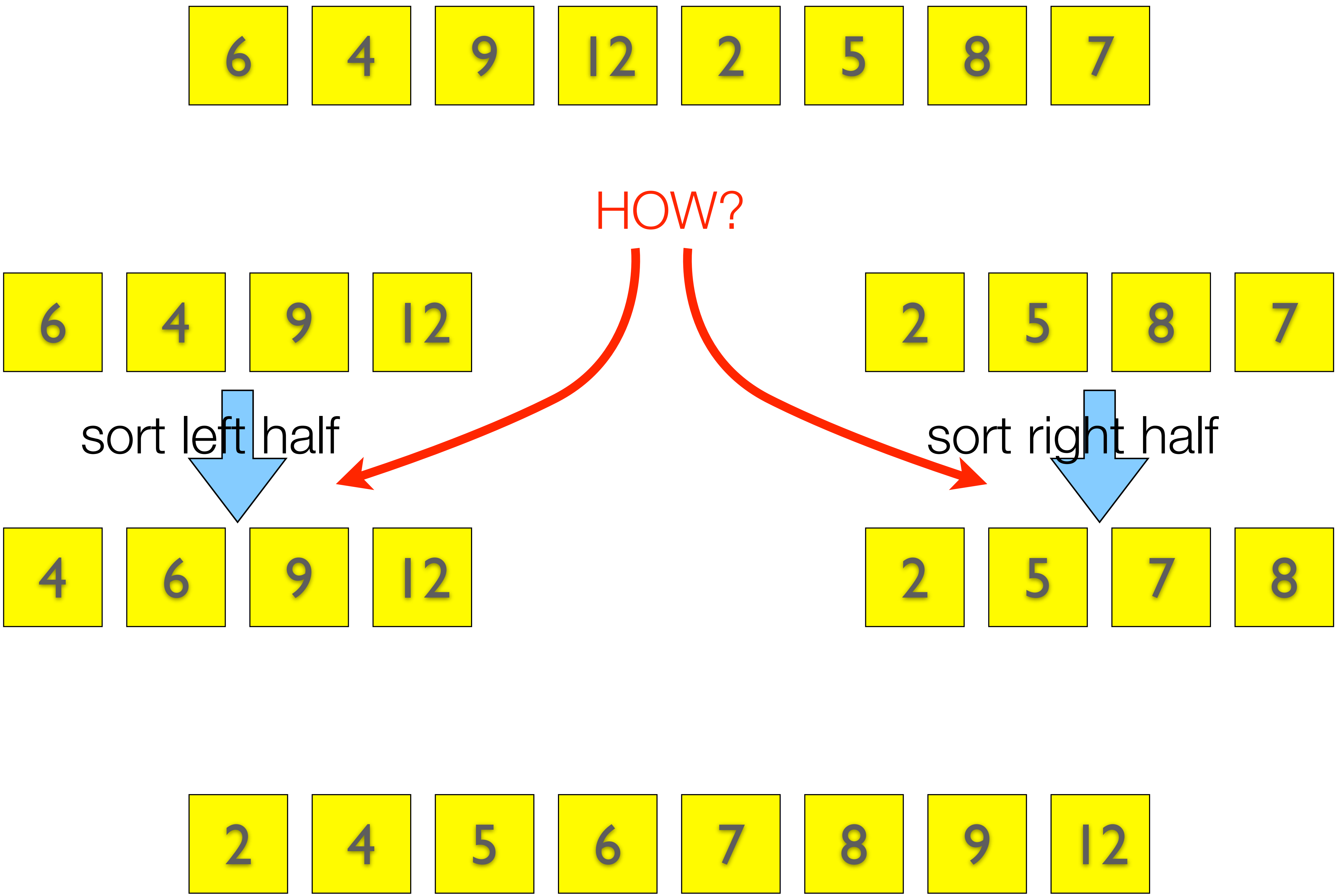
mergesort



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)

1. if (start < end)
2. $q \leftarrow \left\lfloor \frac{end - start}{2} \right\rfloor$
3. sort (A, start, q)
4. sort (A, q+1, end)
5. merge two halves
6. else ....

mergesort(A,start,end)

1 `if start < end` — 1

2 $q \leftarrow \lfloor (\text{start} + \text{end})/2 \rfloor$ — 1

3 `mergesort(A,start,q)` — $T(\lceil \frac{n}{2} \rceil)$
`mergesort(A,q+1,end)` — $T(\lceil \frac{n}{2} \rceil)$

4 `merge(A,start,q,end)` — $O(n)$

5 `else ...`

$$T(n) = 2T(\lceil \tfrac{n}{2} \rceil) + O(n)$$

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 ← 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

mergesort(`A,start,end`)

running time?

```
❶ if start < end
```

❷ $q \leftarrow \lfloor (\text{start} + \text{end})/2 \rfloor$

```
❸     mergesort(A,start,q)
      mergesort(A,q+1,end)

❹     merge(A,start,q,end)

❺  else ...
```

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

show: $T(n) = O(n \log n)$. Prove $T(n) \leq n \log n$

→ Observe that $T(2) \leq 2 \cdot \log 2$, and so on for the first $2, 3, 4 \ldots n_0$ numbers.

Consider $T(n_0+1)$. We know $T(n_0+1) = 2\,T\left(\frac{n_0+1}{2}\right) + (n_0+1)$

∘ But now we can use the fact that $\frac{n_0+1}{2} \leq n_0$ to conclude that

$$T(n_0+1) \leq 2 \cdot \left(\frac{n_0+1}{2}\right) \cdot \log\left(\frac{n_0+1}{2}\right) + (n_0+1)$$

$$= (n_0+1) \cdot \log\left(\frac{n_0+1}{2}\right) + (n_0+1)$$

$$= (n_0+1)\left[\log(n_0+1) - \log(2)\right] + (n_0+1)$$

$$= (n_0+1)(\log(n_0+1))$$

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

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:

$$T(n) = 3T(n/2) + 8O(n)$$

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

$$T(n) = 3T(n/2) + 8O(n)$$ (guess +chk)

goal: Prove $T(n) = O(n^{\log_2 3})$. More specifically,

Prove $T(n) = n^{\log_2 3} - 16n$

Base case: $T(n) < n^{\log_2 3} - 16n$ for small values of $n < n_0$. (By inspection)

Spse $T(n) < n^{\log_2 3} - 16n$ holds for $n < n_0$.

Consider: $T(n_0+1) = 3T\left(\frac{n_0+1}{2}\right) + 8(n_0+1)$

(this argument is $< n_0$, so the hypothesis/base case applies.)

$$< 3\left[\left(\frac{n_0+1}{2}\right)^{\log_2 3} - 16\left(\frac{n_0+1}{2}\right)\right] + 8(n_0+1)$$

$$< (n_0+1)^{\log_2 3} - 16(n_0+1) - 8(n_0+1) + 8(n_0+1)$$