



CS50









4G85

8BB12
D9HXT



4G85

8BB12
D9HXT





T42C
N7426

4G85

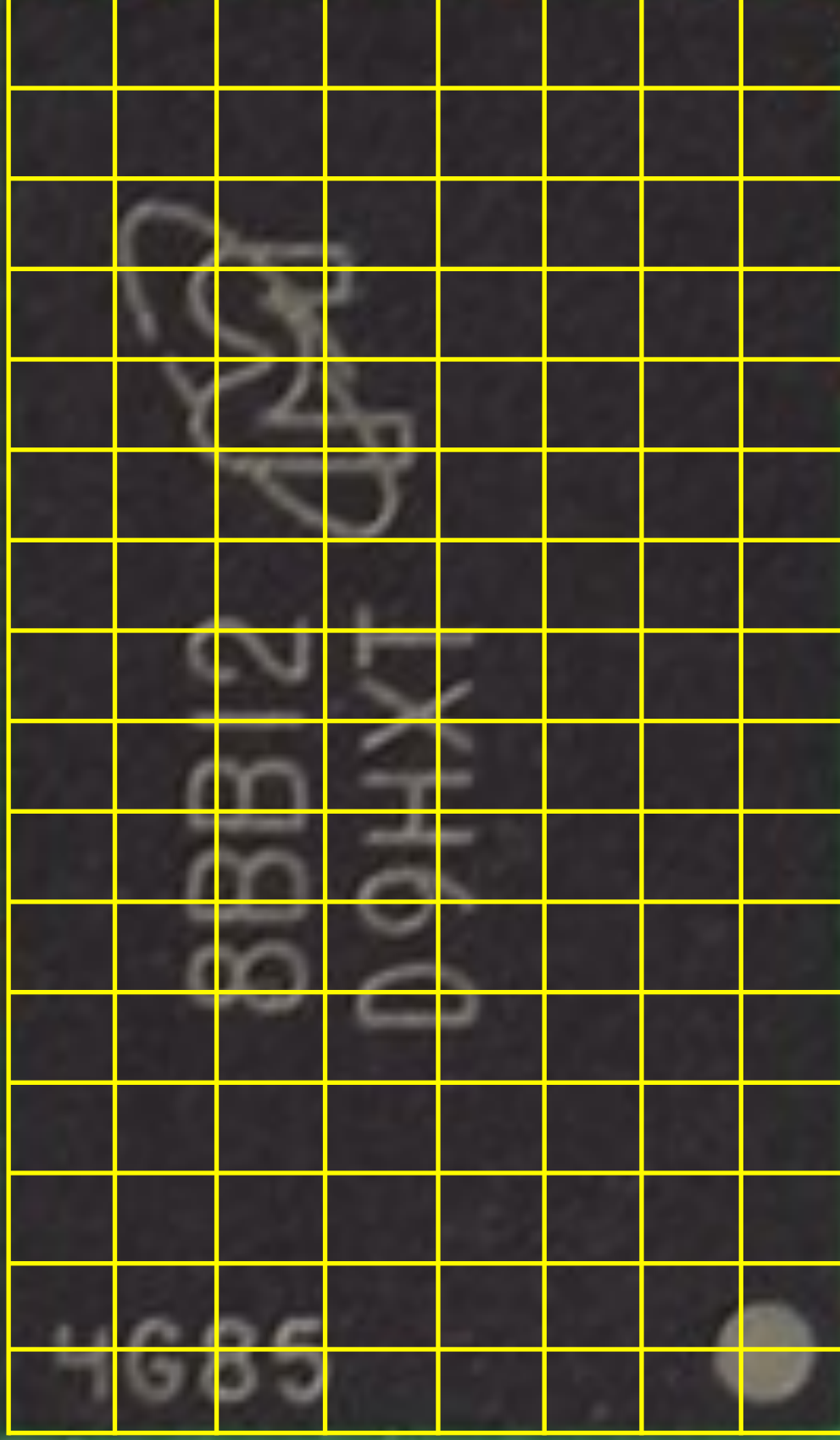
8BB12
D9HXT

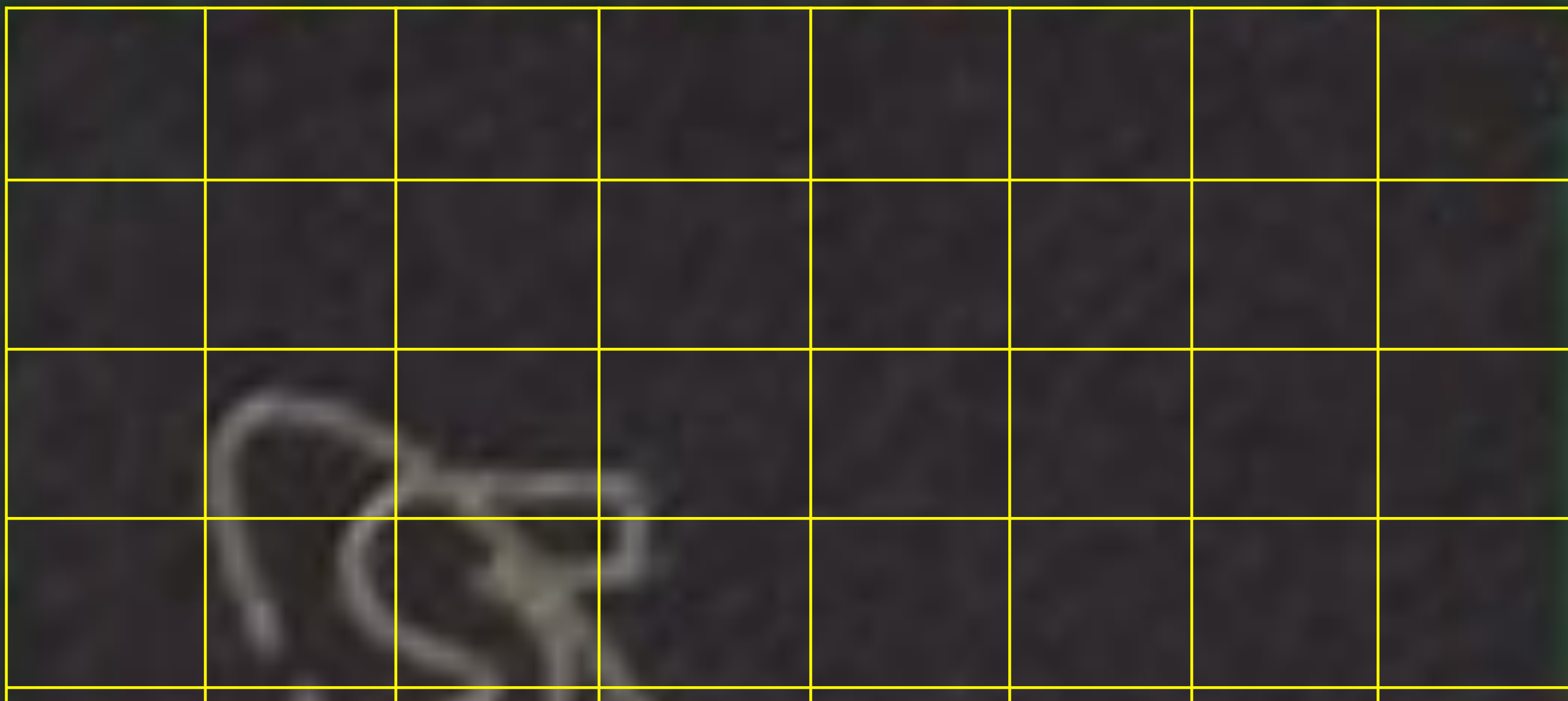


4G85

8BB12
D9HXT







0	1	2	3	4	5	6	7

0 S	1 t	2 e	3 l	4 i	5 o	6 s	7

⁰ S	¹ t	² e	³ l	⁴ i	⁵ o	⁶ s	⁷ \0

0	S	1	t	2	e	3	l	4	i	5	o	6	s	7	\0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----

s	t	e	l	i	o	s	\0
---	---	---	---	---	---	---	----

--	--	--	--	--	--	--	--



linear search

```
for each element in array
    if element you're looking for
        return true
return false
```

binary search

```
look at middle of sorted array
if element you're looking for
    return true
else if element is to left
    search left half of array
else if element is to right
    search right half of array
else
    return false
```

Examination Book



Name _____

Subject _____

Instructor _____

Section _____ Class _____

Date _____ Book No. _____

PONTIAC PAPER CO.

"Home of the Blue Books"

777 Henderson Blvd., Folcroft, PA 19032-0193
(610) 583-5505 FAX (610) 583-4927

4

2

7

5

6

8

3

1

bubble sort

```
repeat until no swaps
  for i from 0 to n-2
    if i'th and i+1'th elements out of order
      swap them
```

selection sort

```
for i from 0 to n-1  
    find smallest element between i'th and n-1'th  
    swap smallest with i'th element
```

insertion sort

```
for i from 1 to n-1
    call 0'th through i-1'th elements the "sorted side"
    remove i'th element
    insert it into sorted side in order
```

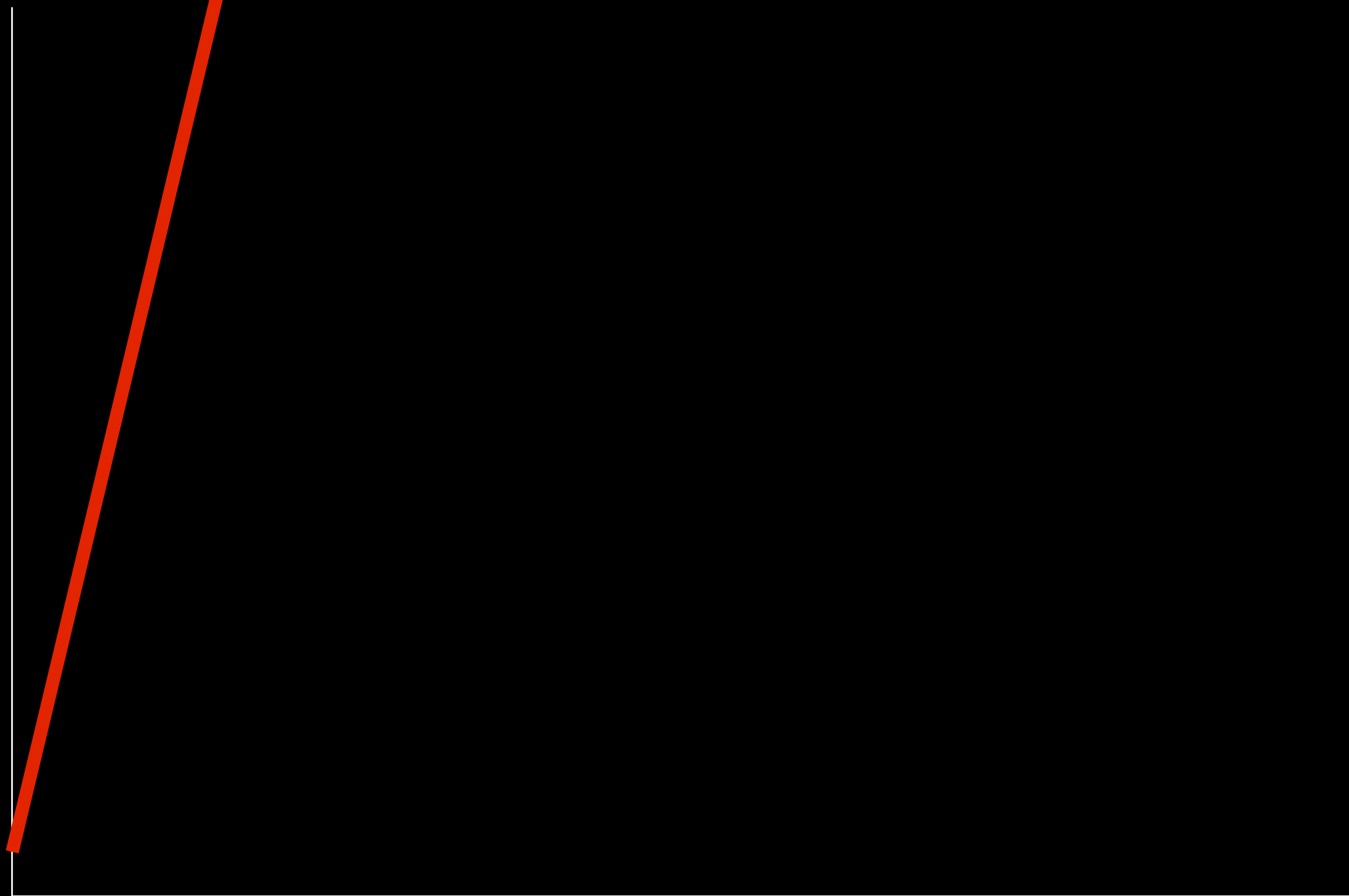
algorithms

running time

time to solve

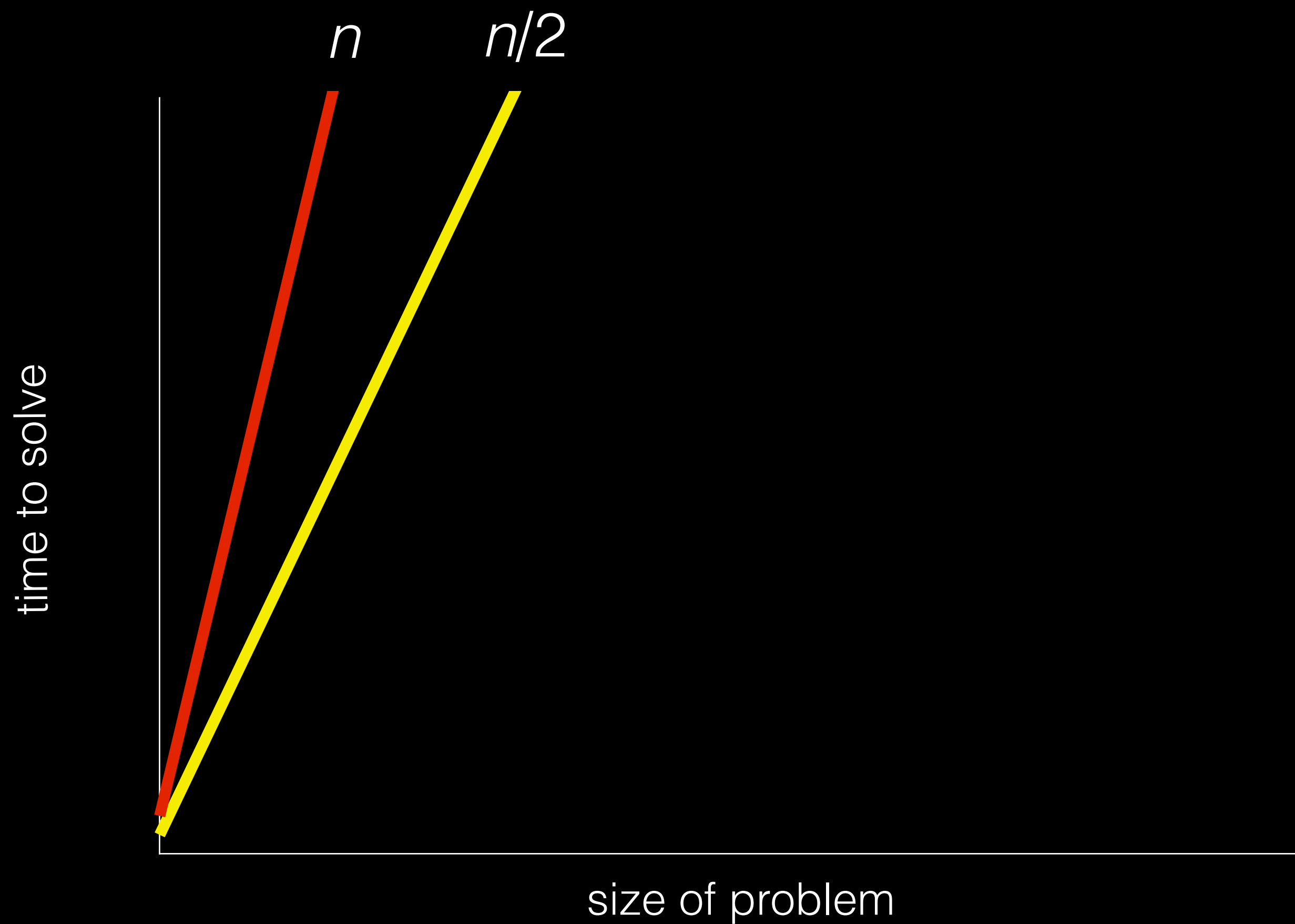
size of problem

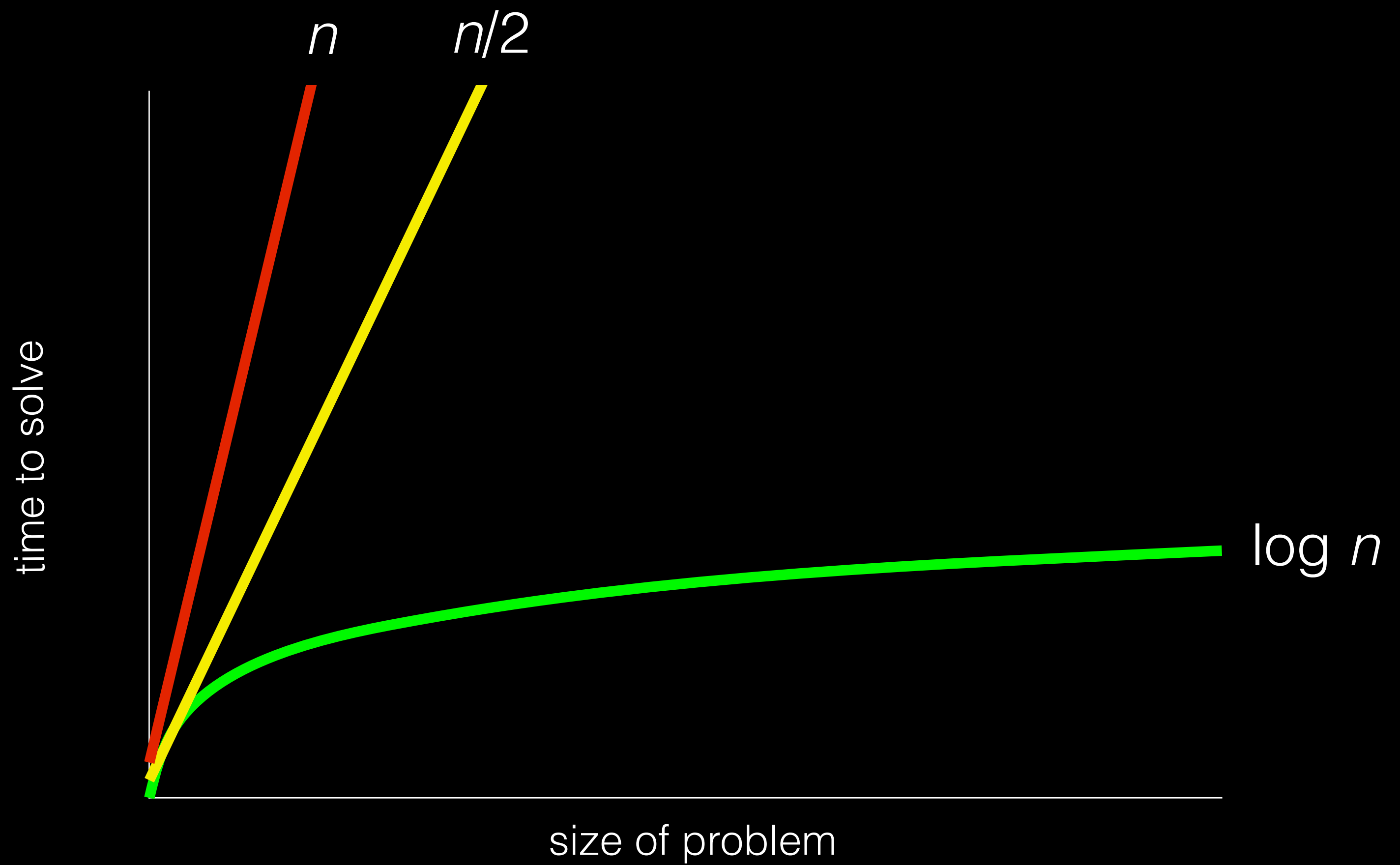
time to solve



n

size of problem





bubble sort

$$(n - 1)$$

$$(n - 1) + (n - 2)$$

$$(n-1) + (n-2) + \dots + 1$$

$$(n-1) + (n-2) + \dots + 1$$

$$n(n-1)/2$$

$$(n - 1) + (n - 2) + \dots + 1$$

$$n(n - 1)/2$$

$$(n^2 - n)/2$$

$$(n - 1) + (n - 2) + \dots + 1$$

$$n(n - 1)/2$$

$$(n^2 - n)/2$$

$$n^2/2 - n/2$$

1,000,000

$$n^2/2 - n/2$$

$$n^2/2 - n/2$$

$$1,000,000^2/2 - 1,000,000/2$$

$$n^2/2 - n/2$$

$$1,000,000^2/2 - 1,000,000/2$$

$$500,000,000,000 - 500,000$$

$$n^2/2 - n/2$$

$$1,000,000^2/2 - 1,000,000/2$$

$$500,000,000,000 - 500,000$$

$$499,999,500,000$$

$$n^2/2 - n/2$$

$$O(n^2)$$

O

$$O(n^2)$$

$$O(n \log n)$$

$$O(n)$$

$$O(\log n)$$

$$O(1)$$

...

$$O(n^2)$$

$$O(n \log n)$$

$$O(n)$$

$$O(\log n)$$

$$O(1)$$

...

$O(n^2)$

$O(n \log n)$

$O(n)$

$O(\log n)$

$O(1)$

...

Ω

$$\Omega(n^2)$$

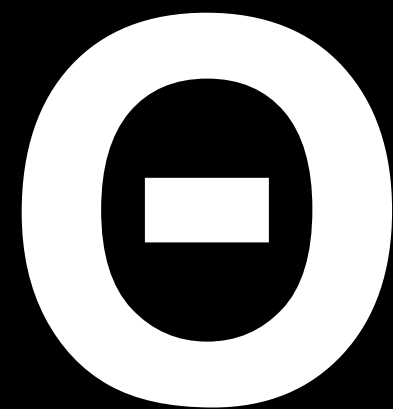
$$\Omega(n \log n)$$

$$\Omega(n)$$

$$\Omega(\log n)$$

$$\Omega(1)$$

...



$$\Theta(n^2)$$

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

$$\Theta(n)$$

$$\Theta(\log n)$$

$$\Theta(1)$$

...

$O(n^2)$

$O(n^2)$

merge sort

```
On input of  $n$  elements
  if  $n < 2$ 
    return
  else
    sort left half of elements
    sort right half of elements
    merge sorted halves
```

4

2

7

5

6

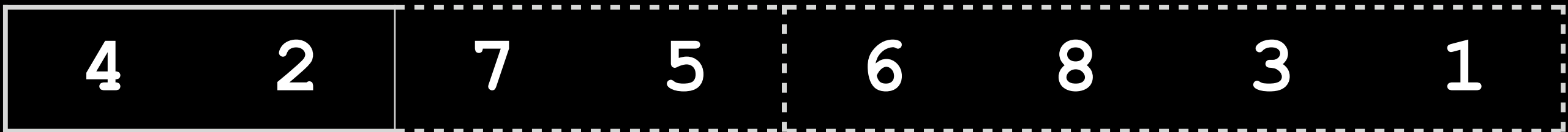
8

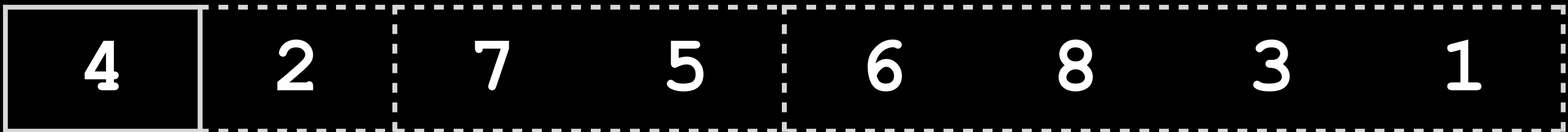
3

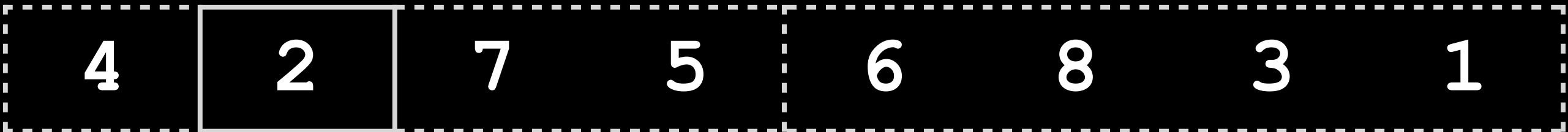
1

4	2	7	5	6	8	3	1
---	---	---	---	---	---	---	---

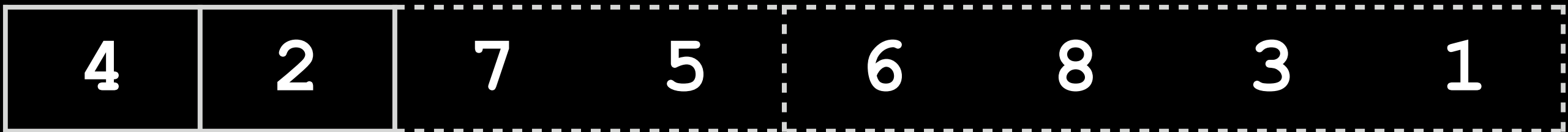
4	2	7	5	6	8	3	1
---	---	---	---	---	---	---	---

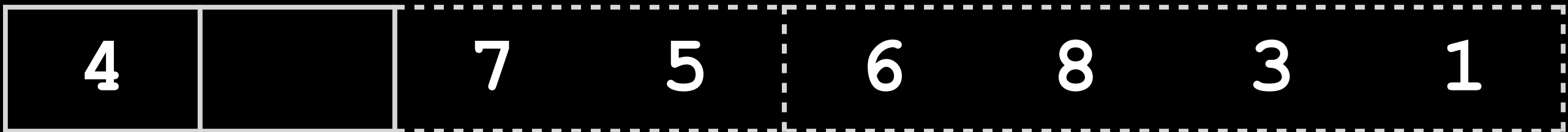


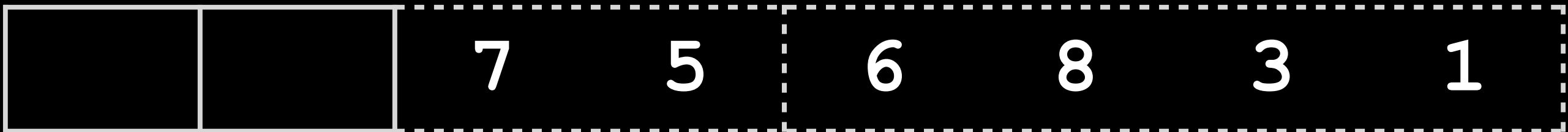


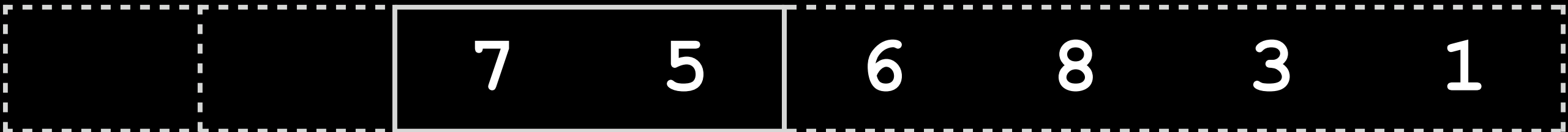


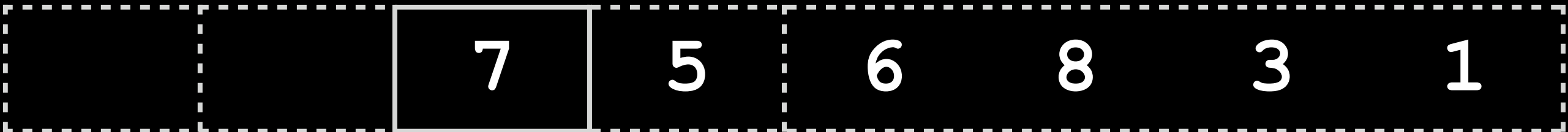
4	2	7	5	6	8	3	1
---	---	---	---	---	---	---	---

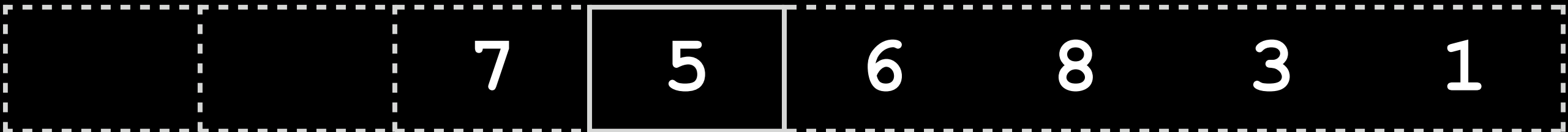


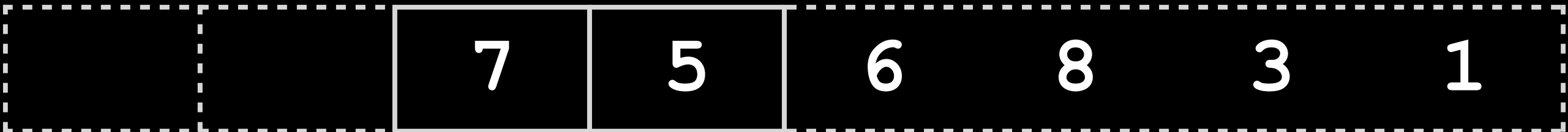


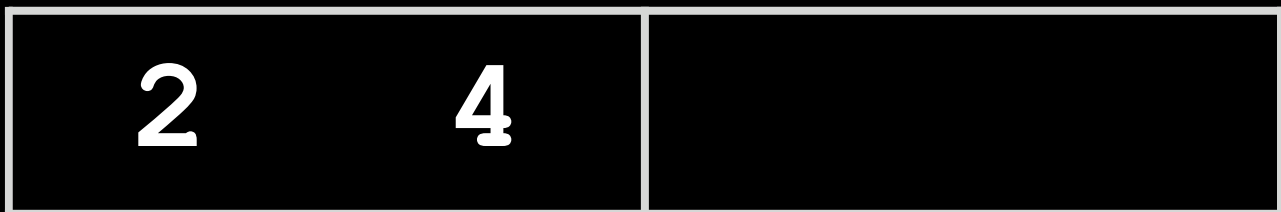
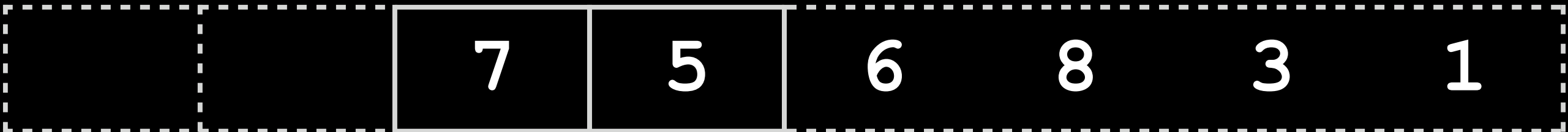


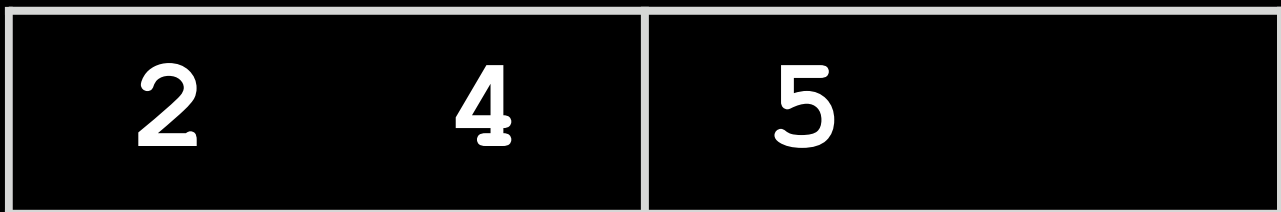
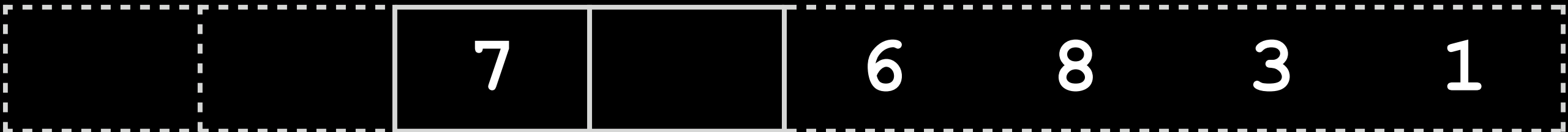


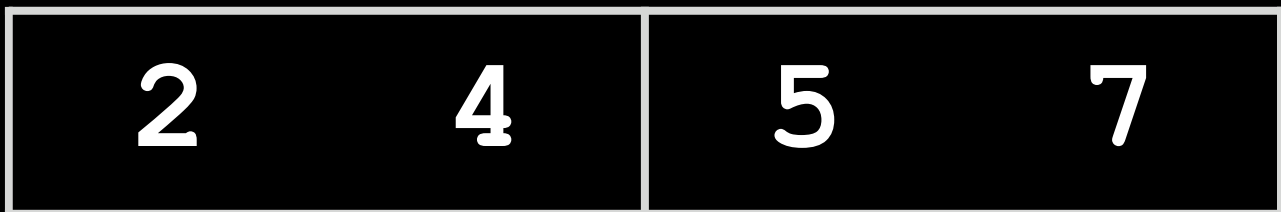
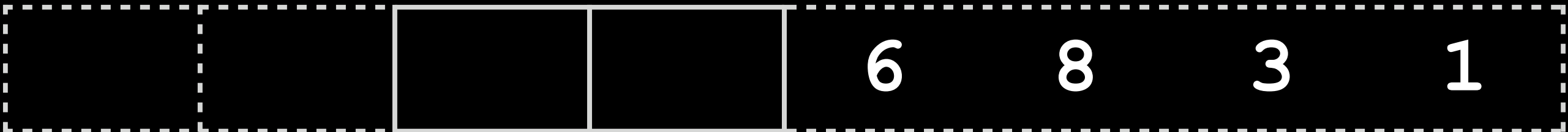












				6	8	3	1
--	--	--	--	---	---	---	---

2	4	5	7
---	---	---	---

--

				6	8	3	1
--	--	--	--	---	---	---	---

4	5	7
---	---	---

2

				6	8	3	1
--	--	--	--	---	---	---	---

	5	7
--	---	---

2	4	
---	---	--

				6	8	3	1
--	--	--	--	---	---	---	---

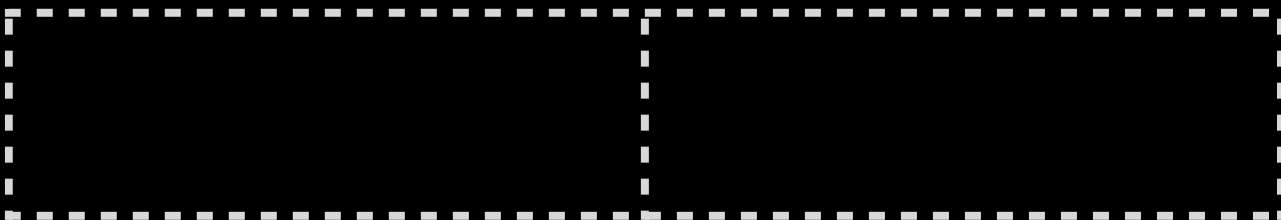
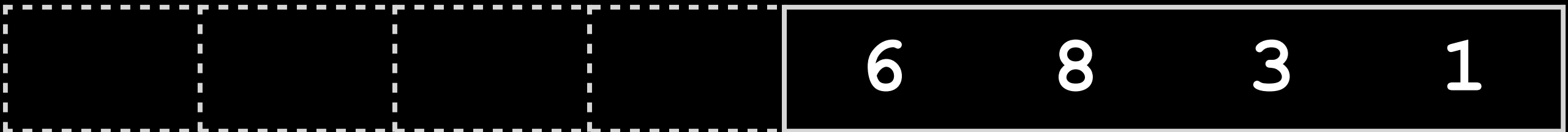
	7
--	---

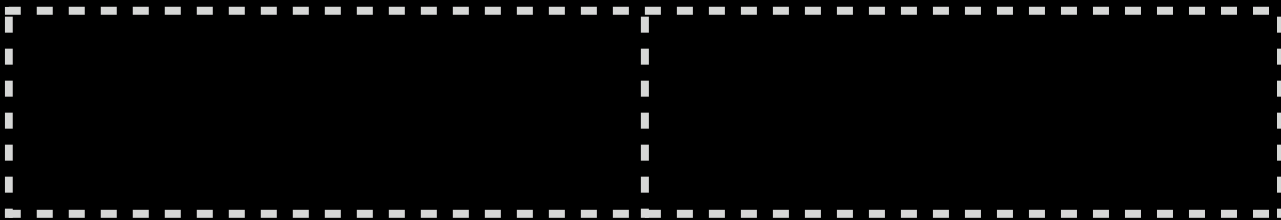
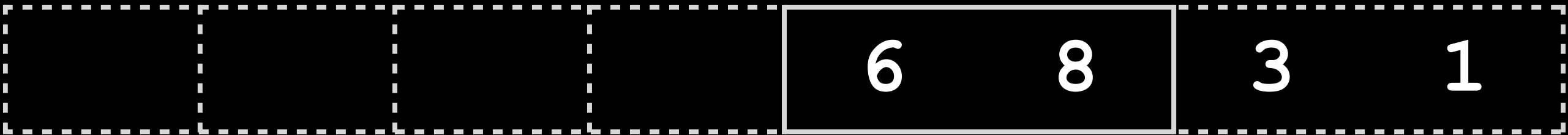
2	4	5	
---	---	---	--

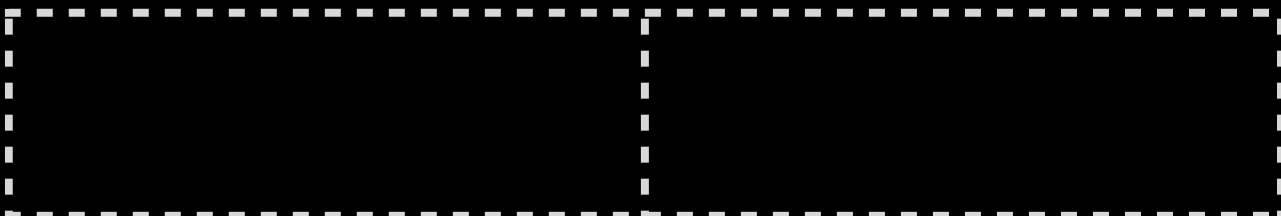
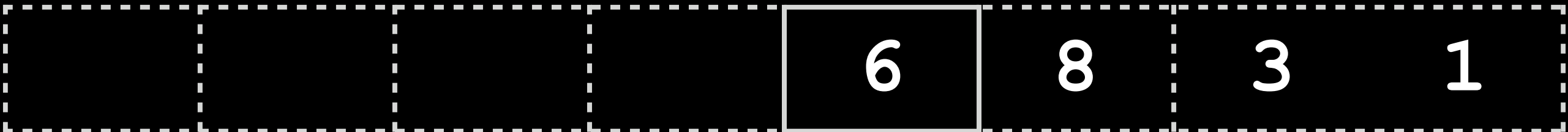
				6	8	3	1
--	--	--	--	---	---	---	---

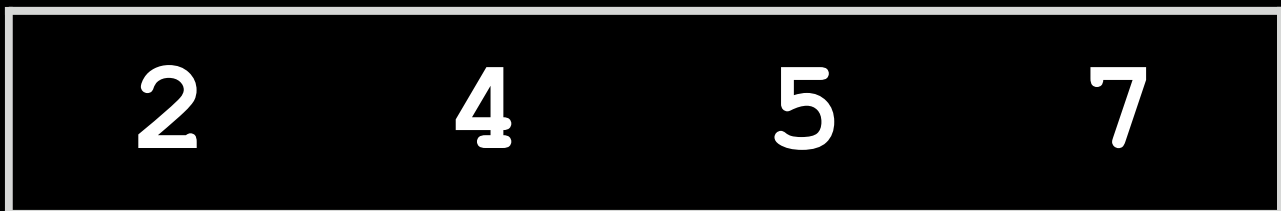
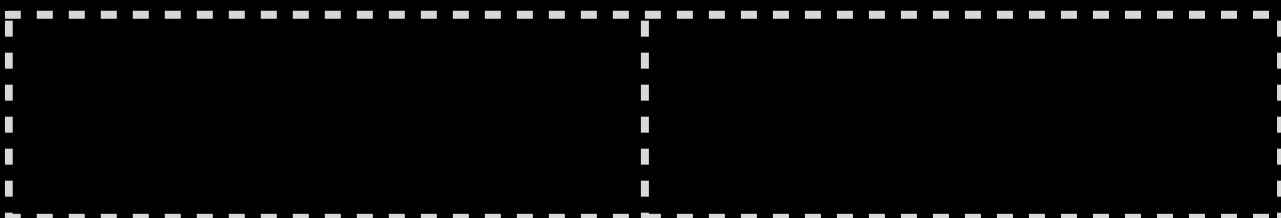
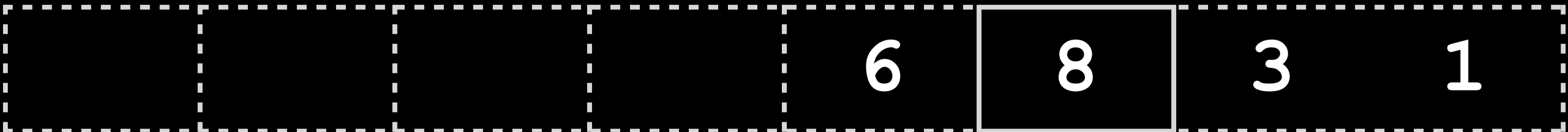
--	--

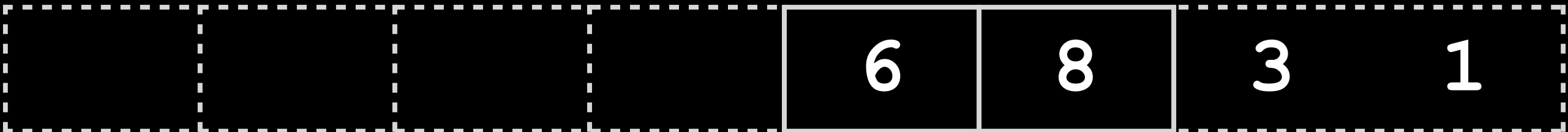
2	4	5	7
---	---	---	---

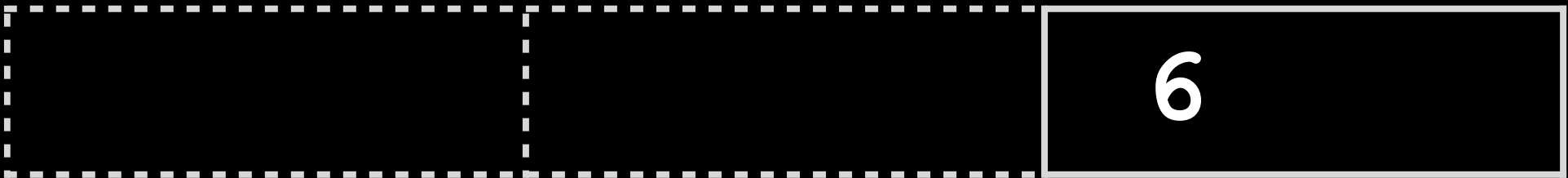
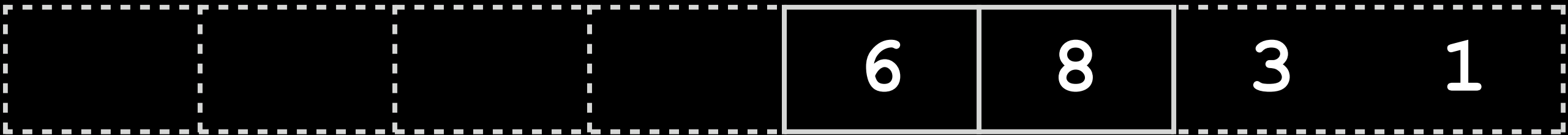


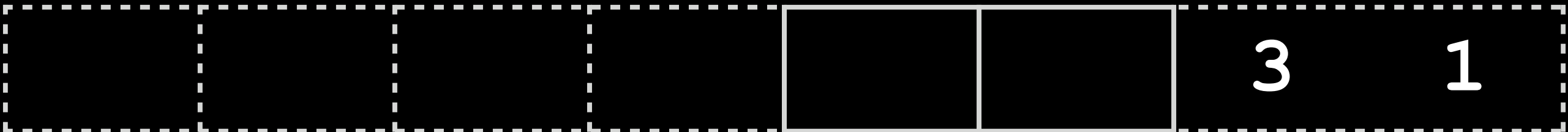


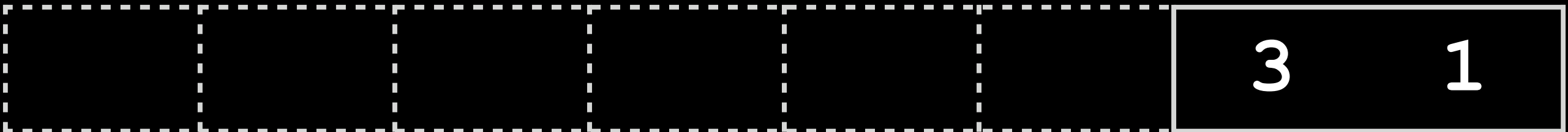


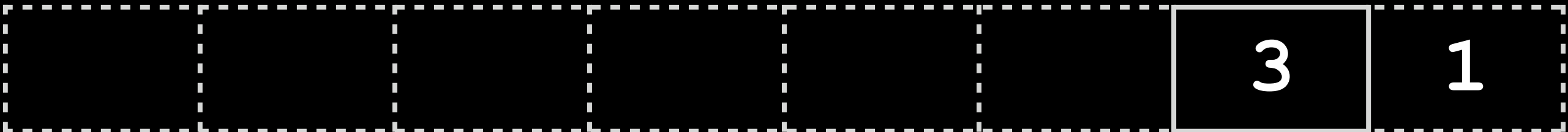


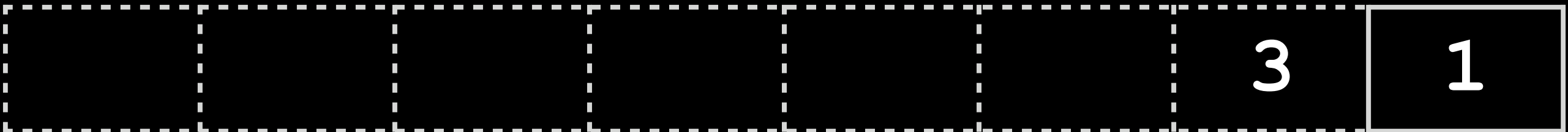


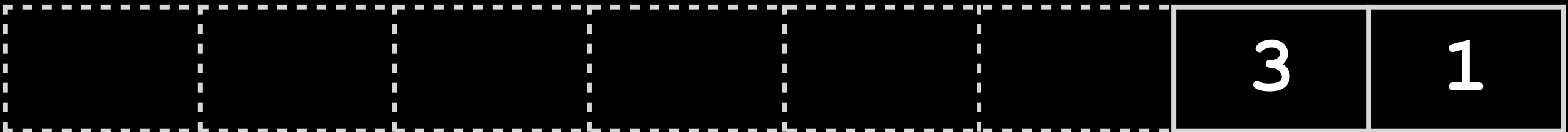


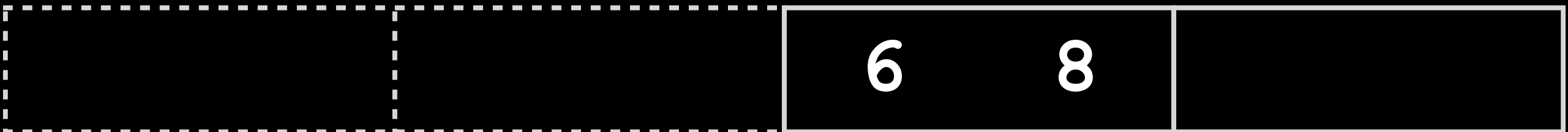
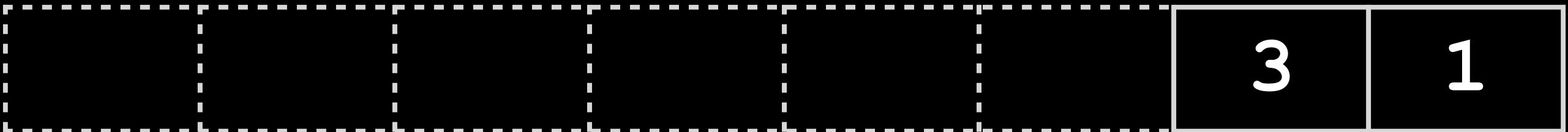


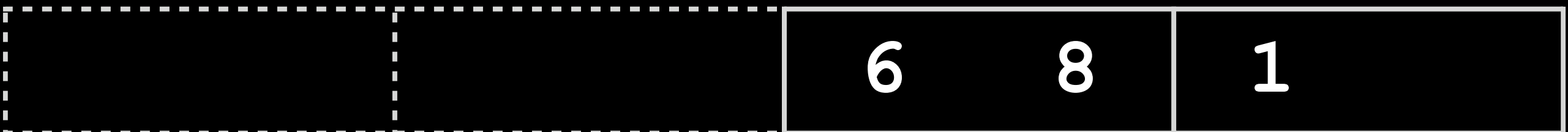
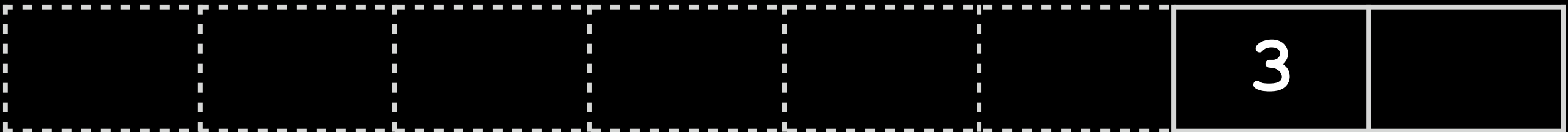


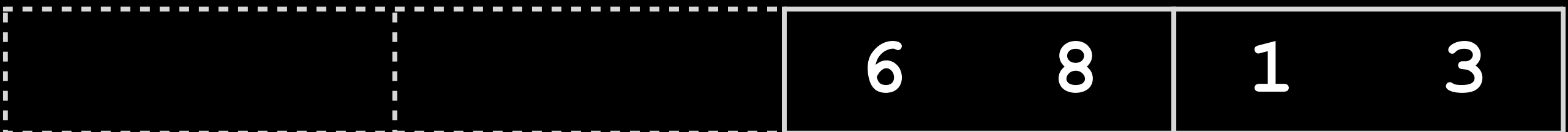
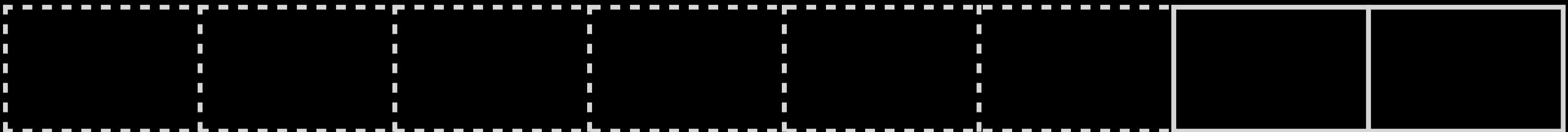












--	--	--	--	--	--	--	--

		6	8	1	3
--	--	---	---	---	---

2	4	5	7		
---	---	---	---	--	--

--	--	--	--	--	--	--	--

		6	8	3
--	--	---	---	---

2	4	5	7	1
---	---	---	---	---

--	--	--	--	--	--	--	--

		6	8	
--	--	---	---	--

2	4	5	7	1	3	
---	---	---	---	---	---	--

--	--	--	--	--	--	--	--

		8	
--	--	---	--

2	4	5	7	1	3	6
---	---	---	---	---	---	---

--	--	--	--	--	--	--	--

--	--	--	--

2	4	5	7	1	3	6	8
---	---	---	---	---	---	---	---

--	--	--	--	--	--	--	--

--	--	--	--

2	4	5	7	1	3	6	8
---	---	---	---	---	---	---	---

--

--	--	--	--	--	--	--	--

--	--	--	--

2	4	5	7	3	6	8
---	---	---	---	---	---	---

1

--	--	--	--	--	--	--	--

--	--	--	--	--	--

4	5	7	3	6	8
---	---	---	---	---	---

1	2				
---	---	--	--	--	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--

4	5	7		6	8
---	---	---	--	---	---

1	2	3					
---	---	---	--	--	--	--	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--

5	7	6	8
---	---	---	---

1	2	3	4				
---	---	---	---	--	--	--	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--

			7			6	8
--	--	--	---	--	--	---	---

1	2	3	4	5			
---	---	---	---	---	--	--	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--

7	8
---	---

1	2	3	4	5	6	
---	---	---	---	---	---	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

							8
--	--	--	--	--	--	--	---

1	2	3	4	5	6	7	
---	---	---	---	---	---	---	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

$O(n \log n)$

On input of n elements

if $n < 2$

return

else

sort left half of elements

sort right half of elements

merge sorted halves

On input of n elements

if $n < 2$

return

else

sort left half of elements

sort right half of elements

merge sorted halves

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

if $n < 2$

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

if $n < 2$

On input of n elements

if $n < 2$

return

else

sort left half of elements

sort right half of elements

merge sorted halves

On input of n elements

if $n < 2$

return

else

sort left half of elements

sort right half of elements

merge sorted halves

On input of n elements

if $n < 2$

return

else

sort left half of elements

sort right half of elements

merge sorted halves

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

if $n \geq 2$

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

if $n \geq 2$

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

if $n \geq 2$

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

if $n \geq 2$

$O(n \log n)$

$O(n \log n)$

On input of n elements

if $n < 2$

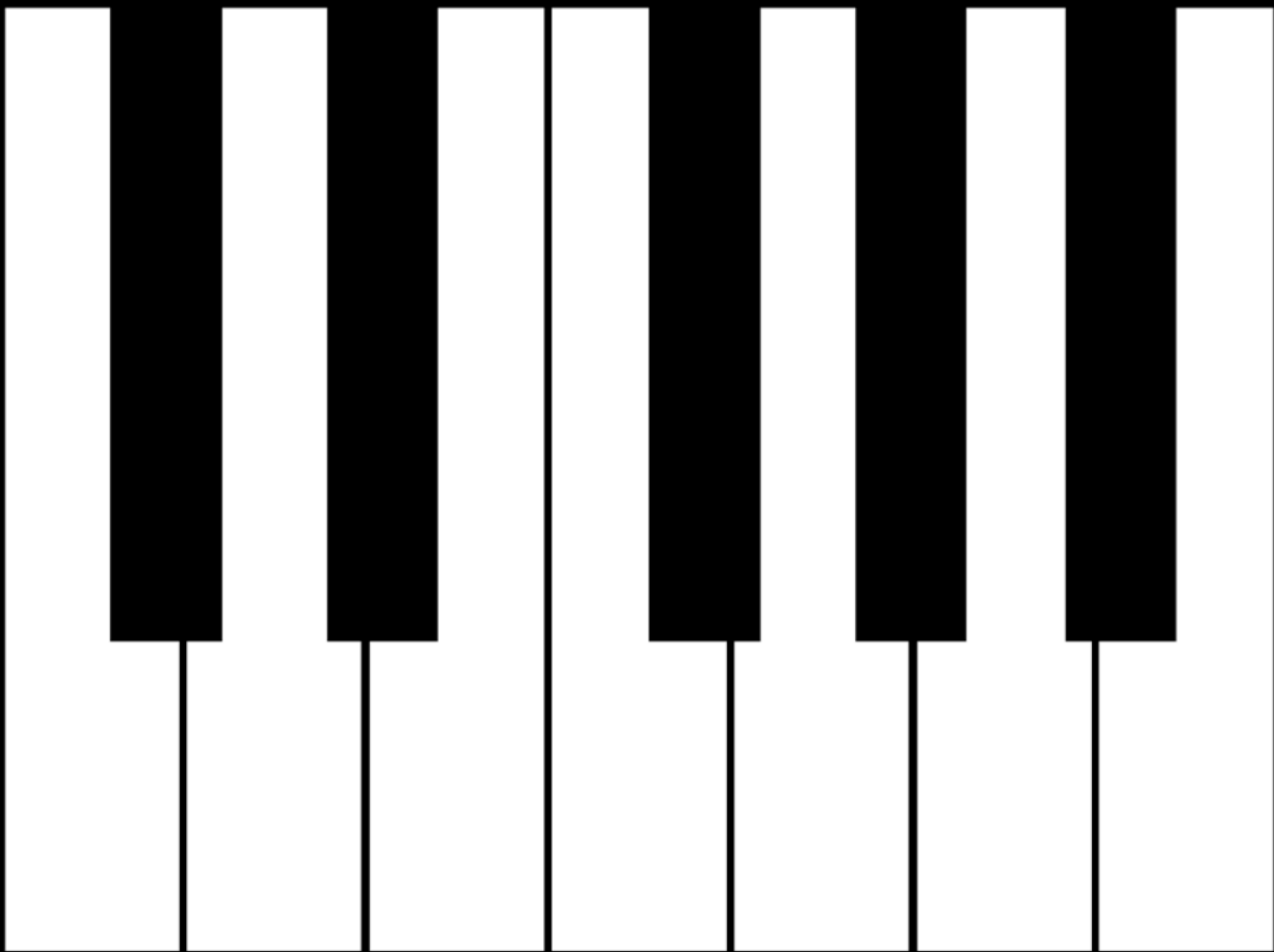
return

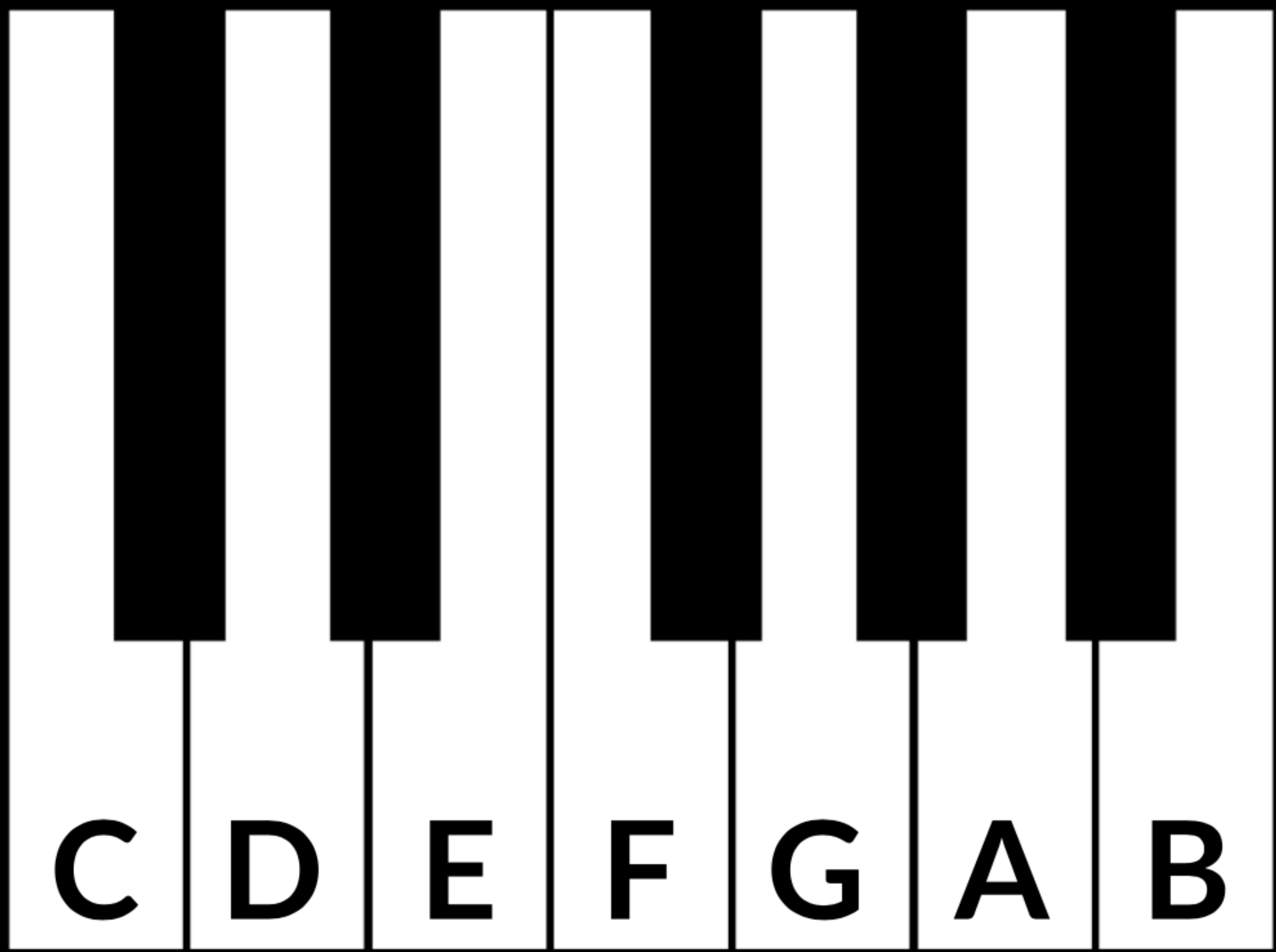
else

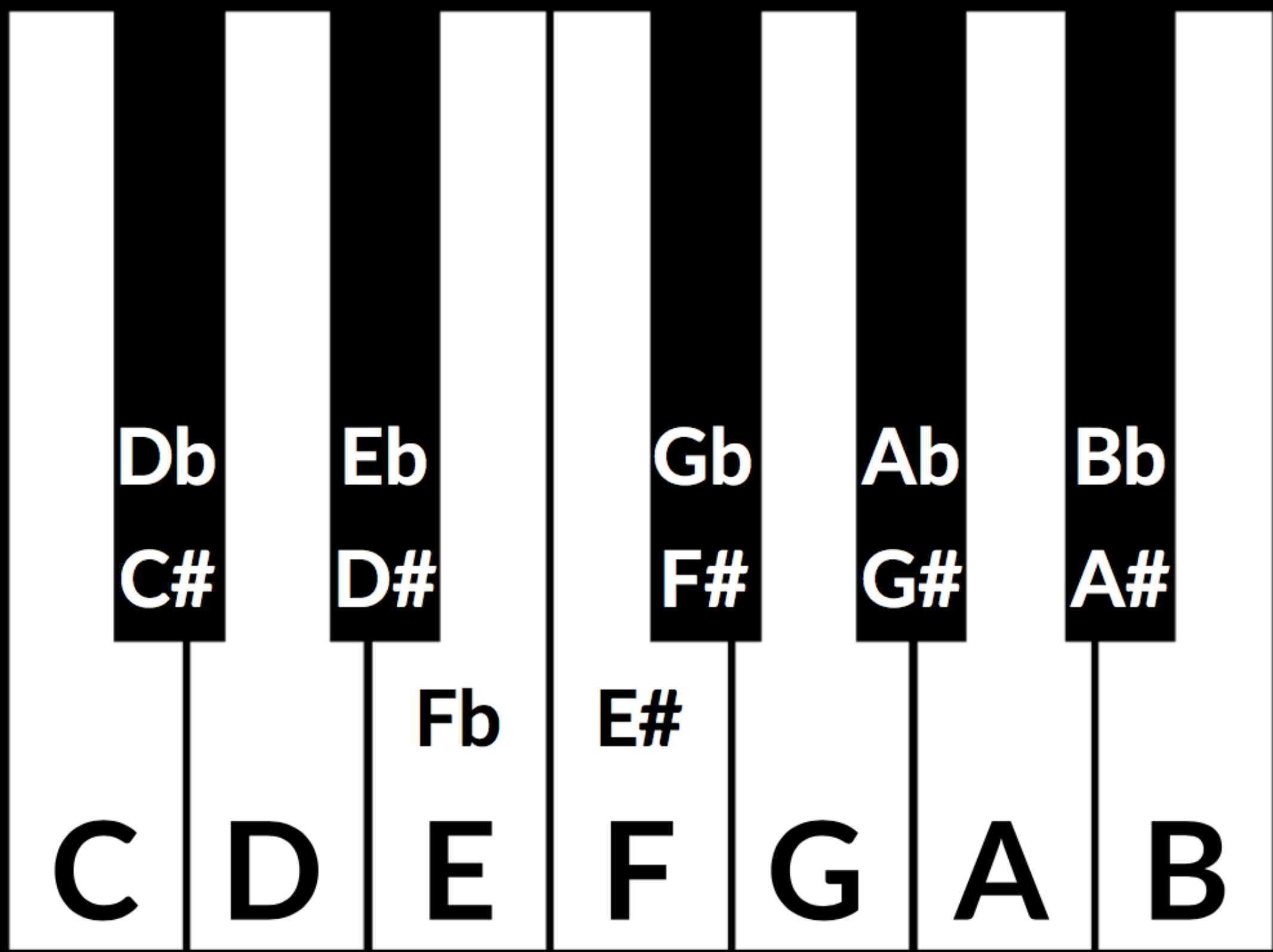
sort left half of elements

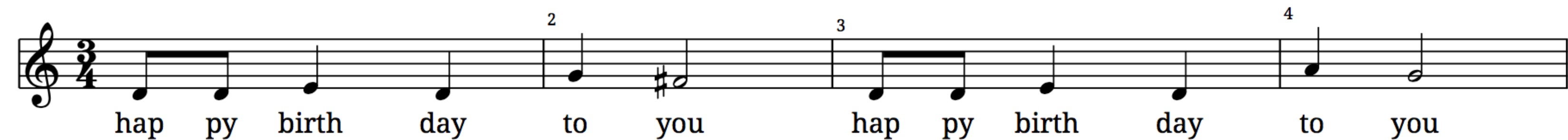
sort right half of elements

merge sorted halves









Insert sort





CS50

