Merge sort

In merge sort, the idea of an algorithm is to sort smaller arrays and then combine those arrays together (merge them) in sorted order.

Merge sort leverages something called **Recursion.**

```
If only one item
    Return
Else
    Sort left half of items
    Sort right half of items
    Merge sorted halves
```

```
If only one item
    Return
Else
    Sort left half of items
    Sort right half of items
    Merge sorted halves
```

7 4 5 2 6 3 8 1

7 4 5 2 6 3 8 1

7 4 5 2 6 3 8 1

7 4 5 2 6 3 8 1

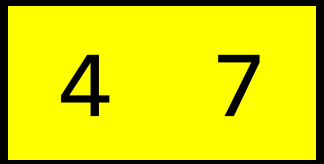
7 4 5 2 6 3 8 1


7 4 5 2 6 3 8 1

7

5 2 6 3 8 1

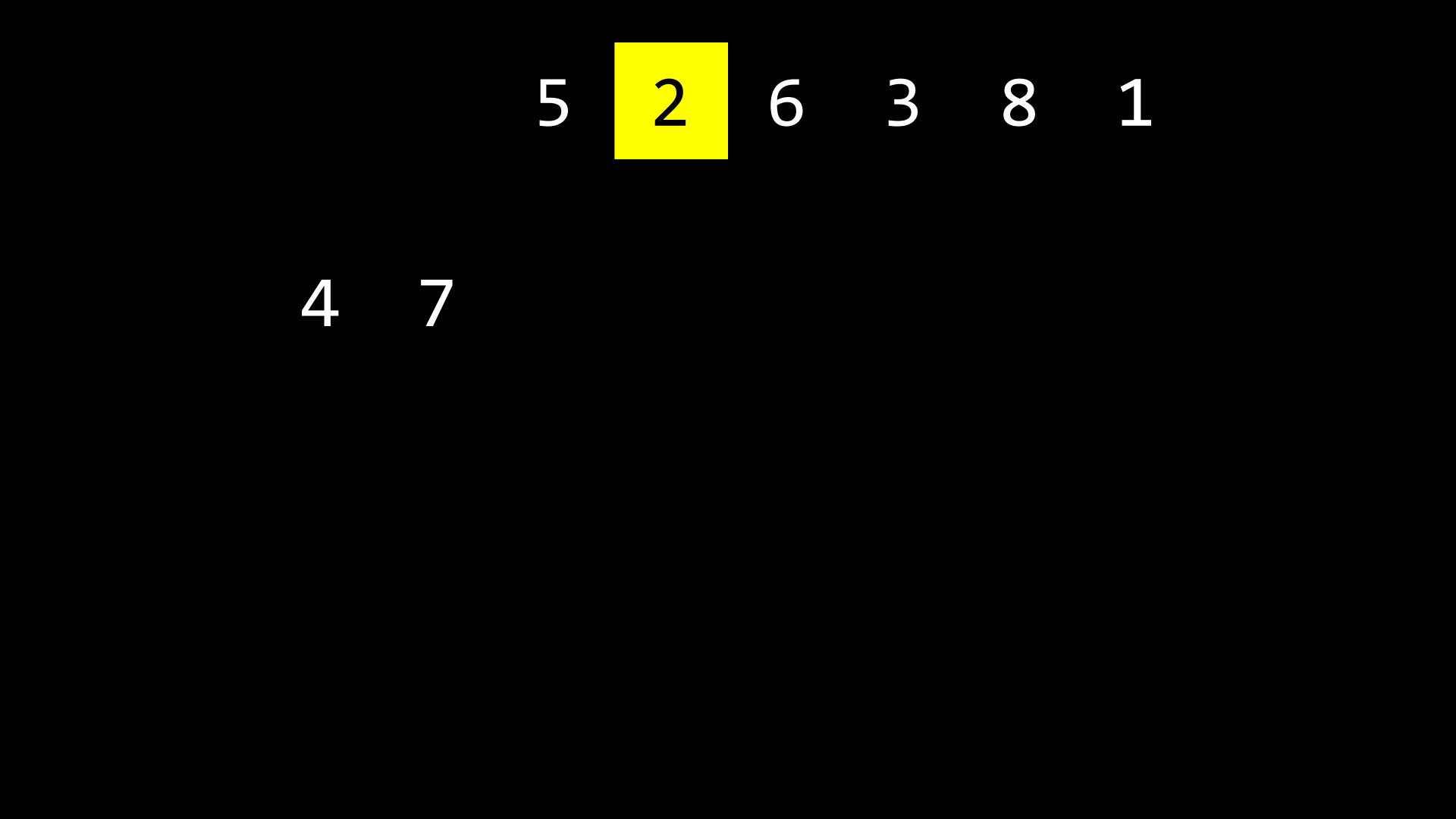
4


5 2 6 3 8 1
4 7

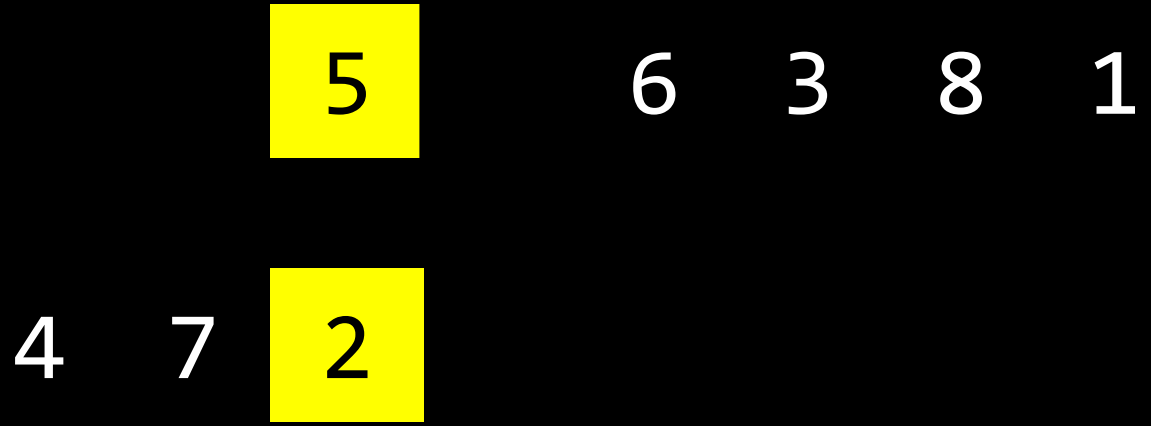
5 2 6 3 8 1

4 7

4 7


5 2 6 3 8 1
4 7

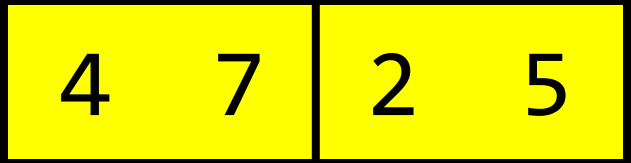
4 7


5
6 3 8 1
4 7 2

6 3 8 1

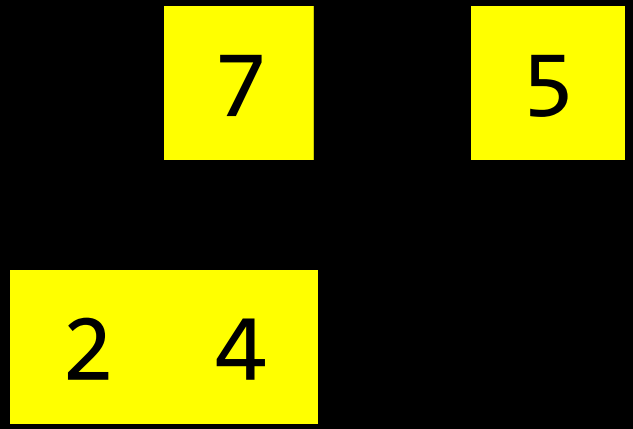
4 7 2 5

6 3 8 1

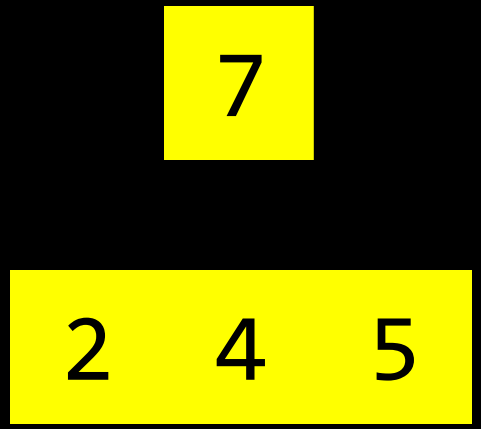


6 3 8 1

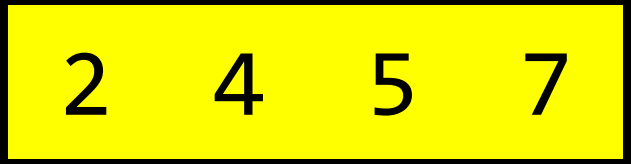
6 3 8 1



6 3 8 1



6 3 8 1



6 3 8 1

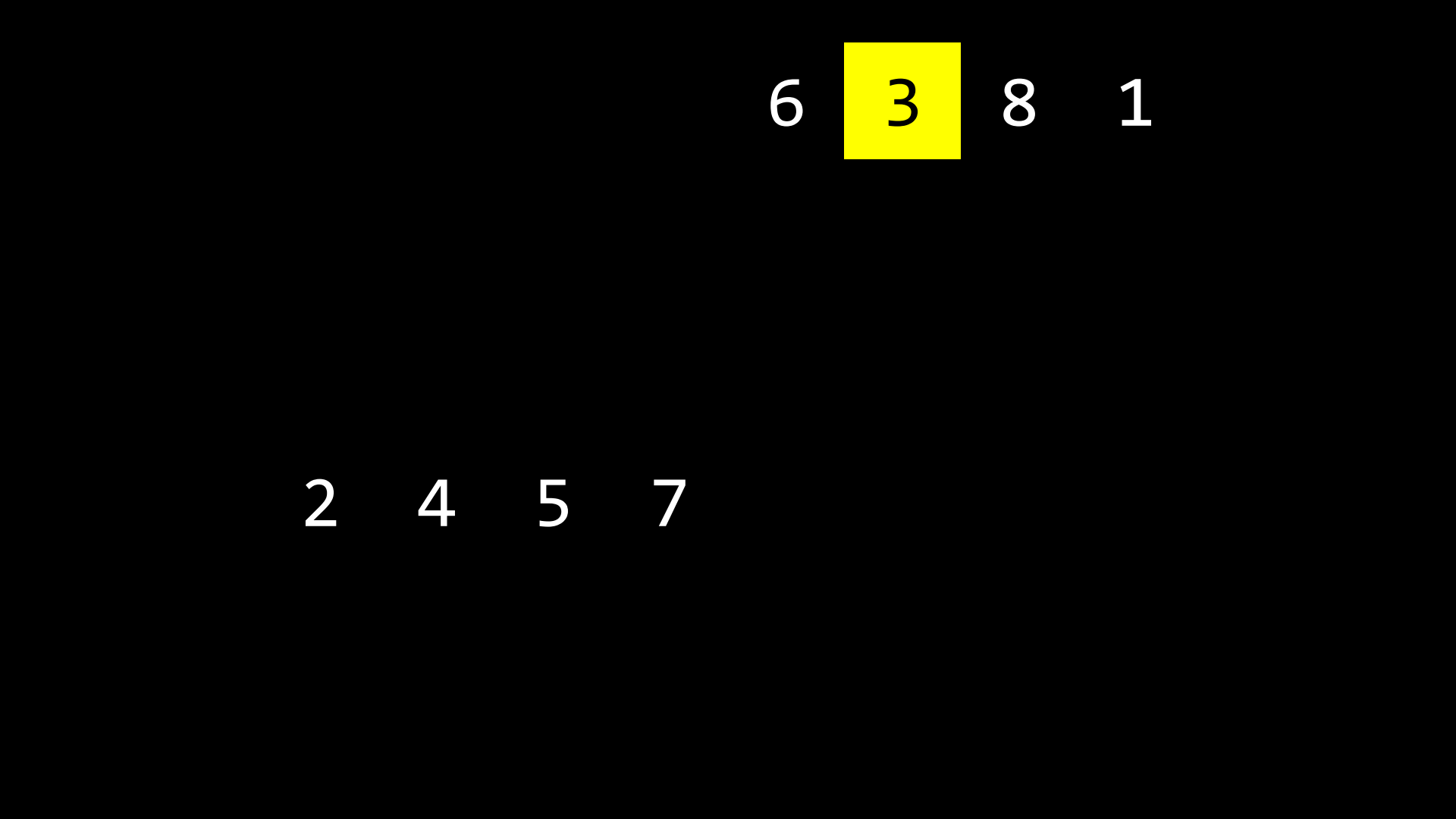
2 4 5 7

6 3 8 1

2 4 5 7

6 3 8 1

2 4 5 7


6 3 8 1
2 4 5 7

2 4 5 7

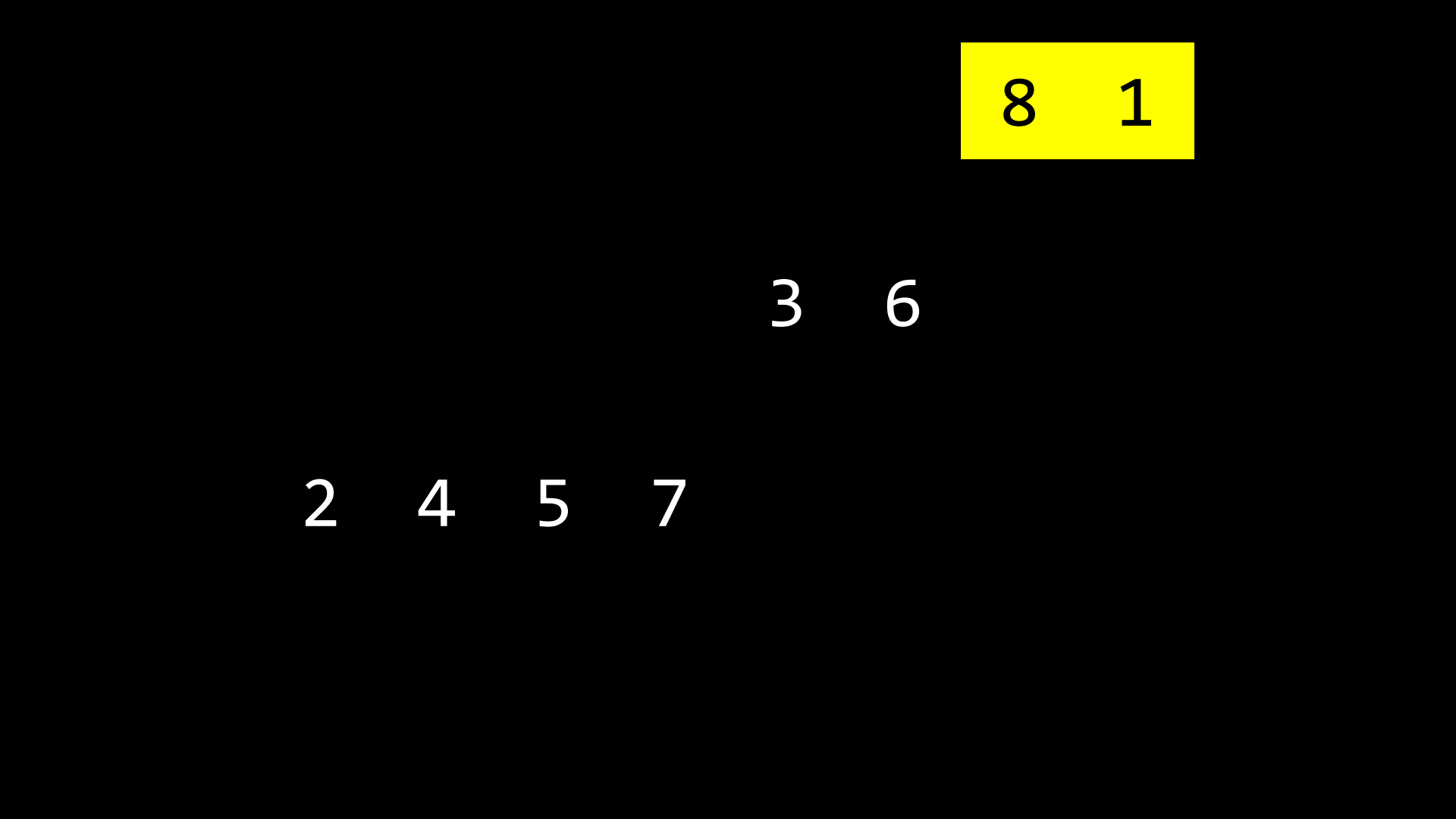
8 1

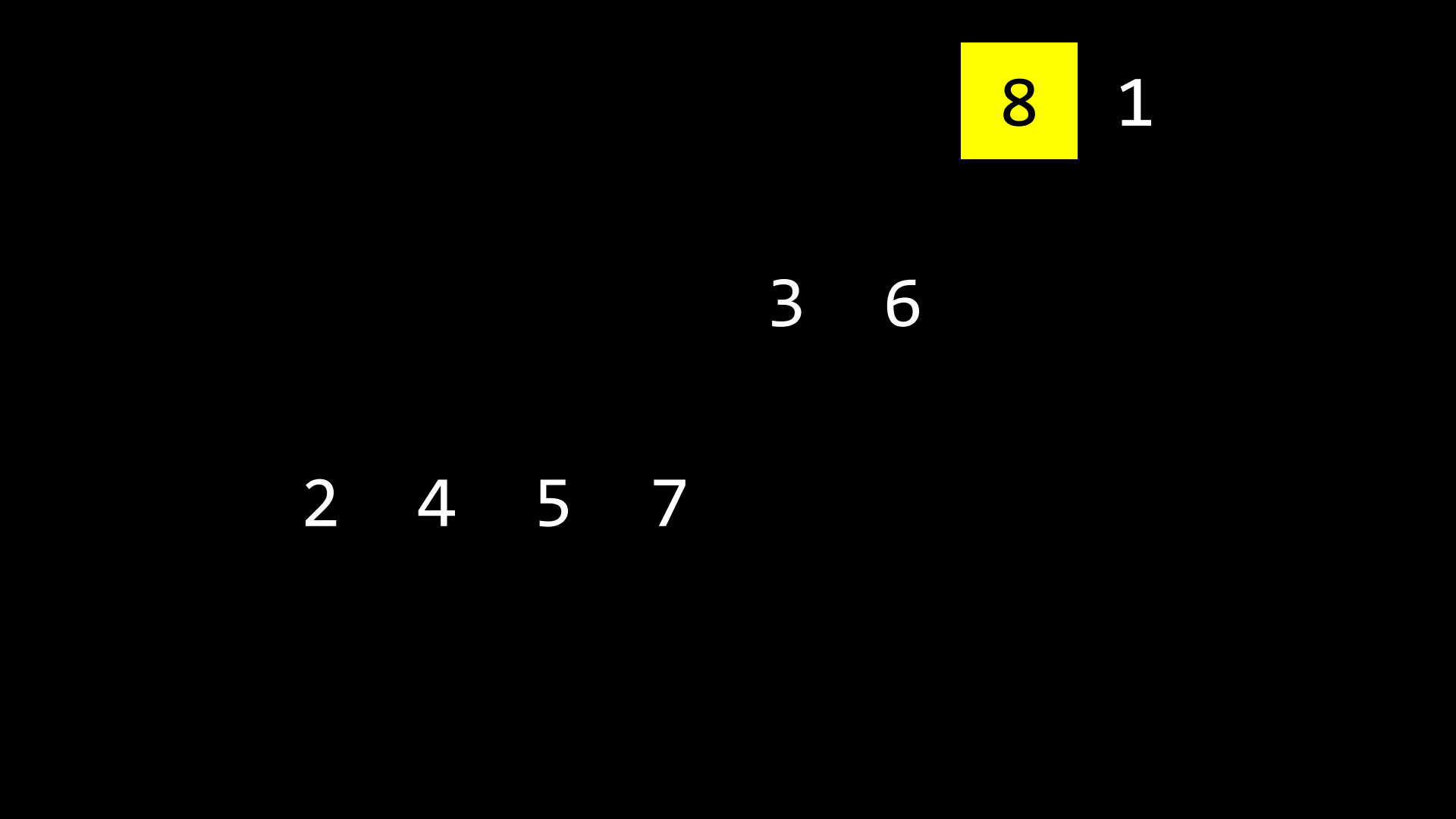
2 4 5 7

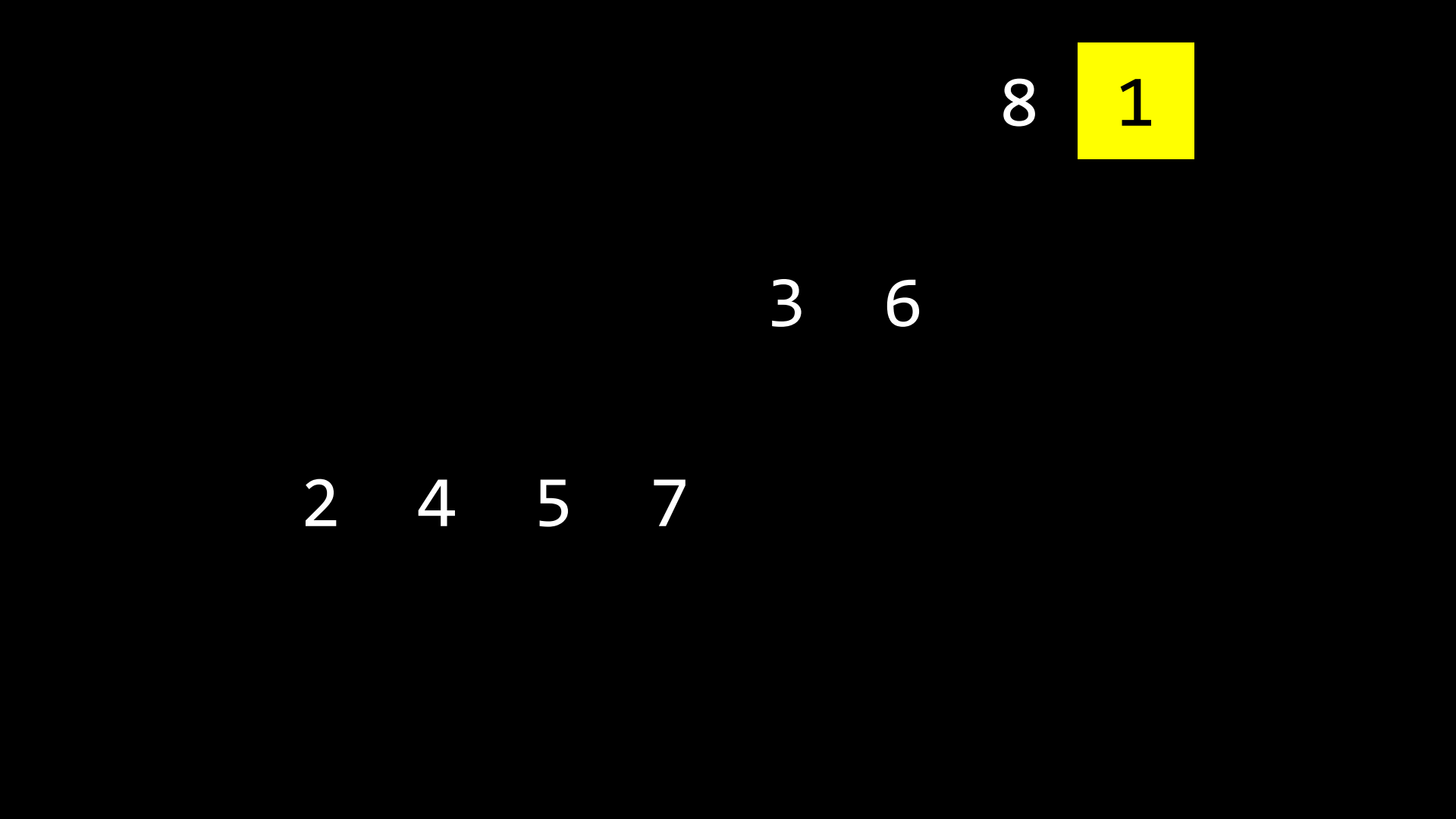
8 1

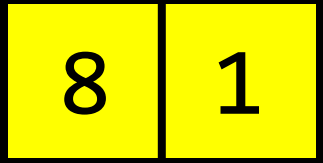
3 6

2 4 5 7


8 1
3 6
2 4 5 7

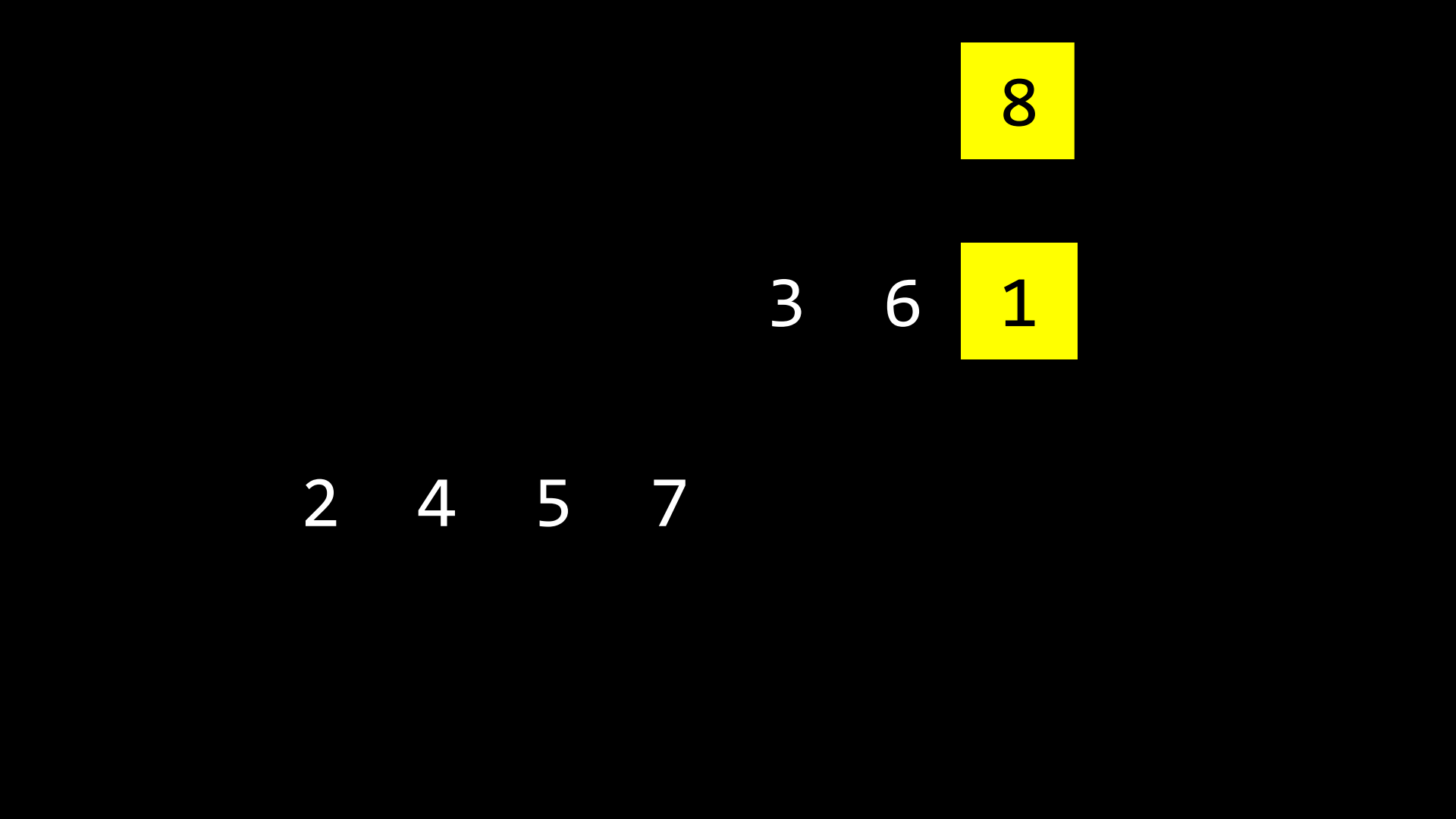

8
1
3
6
2
4
5
7


8
1
3
6
2
4
5
7



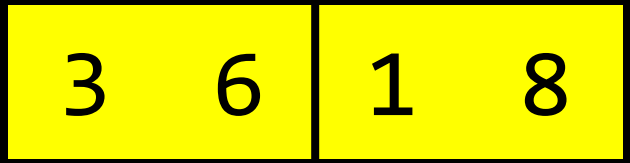
3 6

2 4 5 7

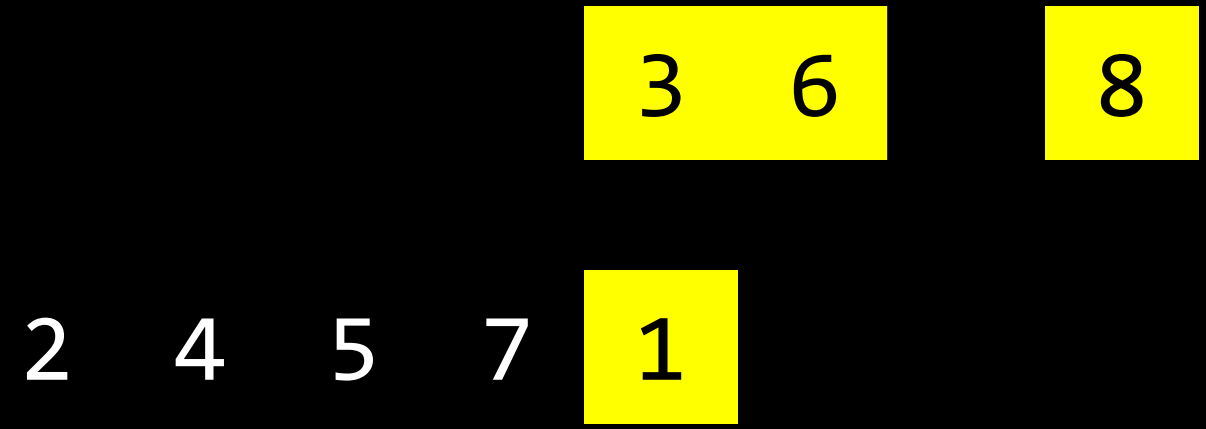

8
3 6 1
2 4 5 7

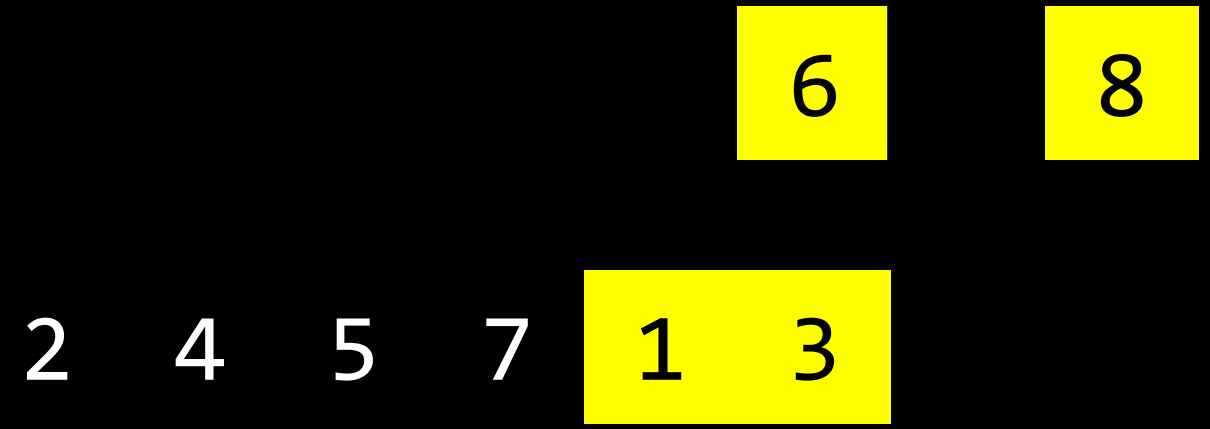
3 6 1 8

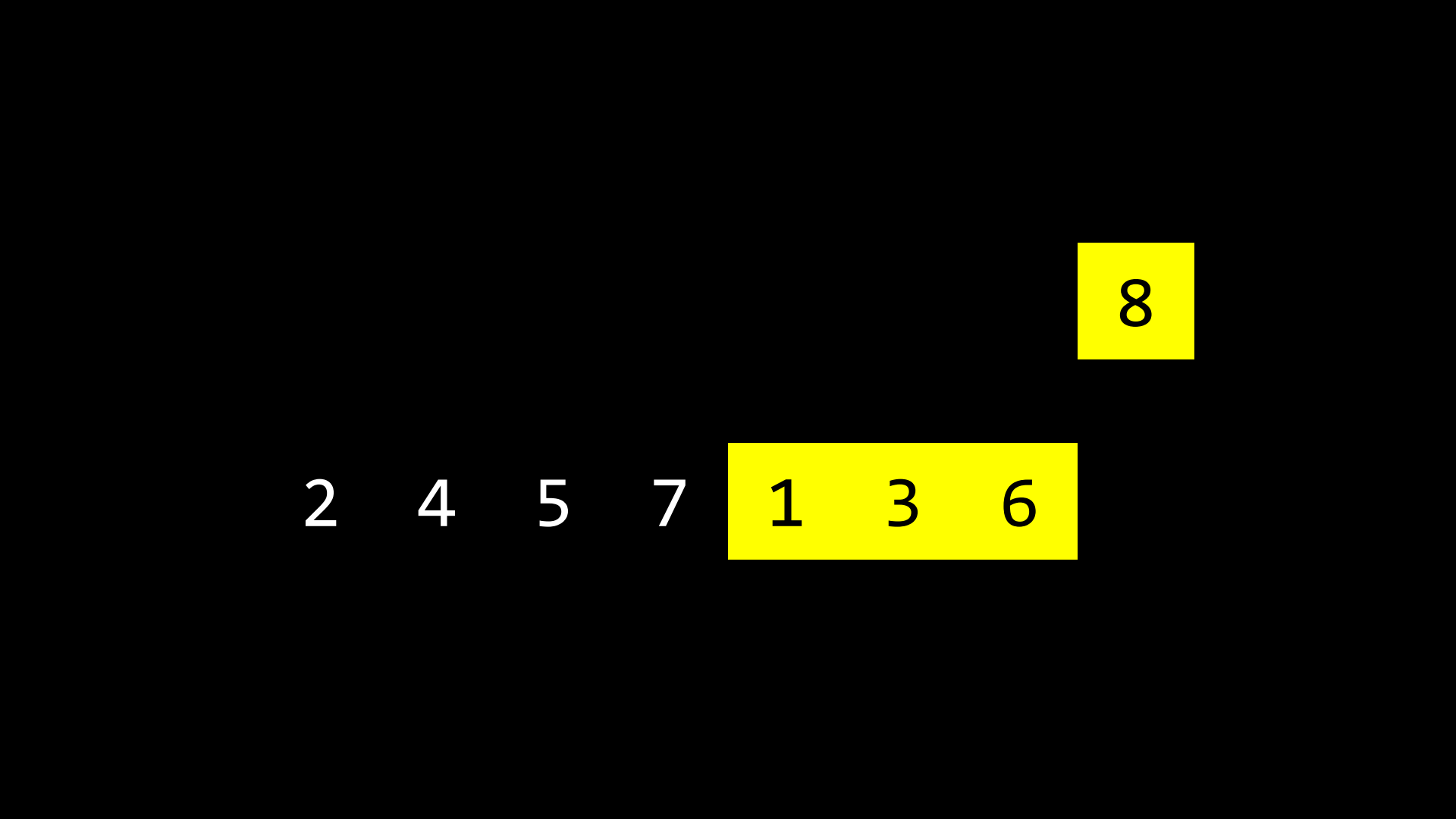
2 4 5 7

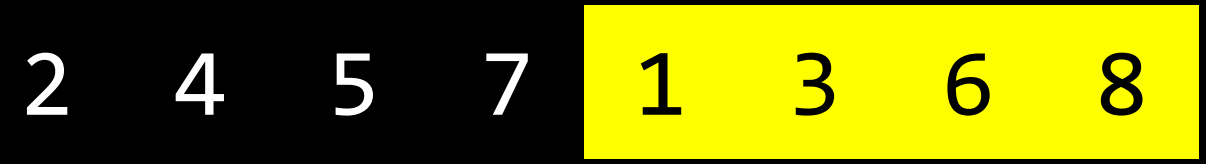


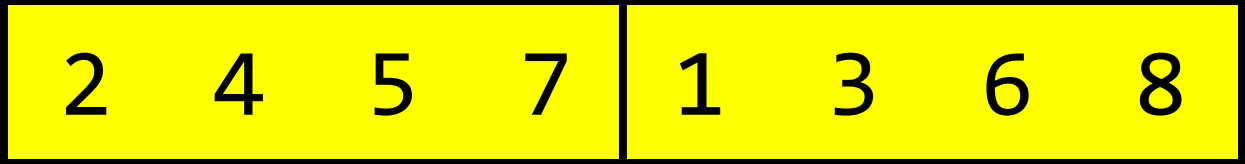
2 4 5 7

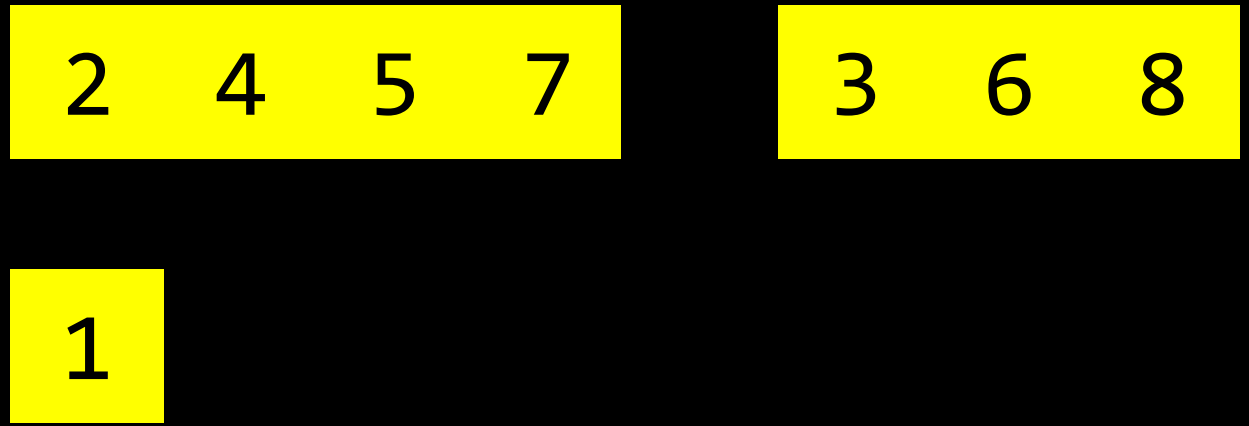

3 6
8
2 4 5 7
1

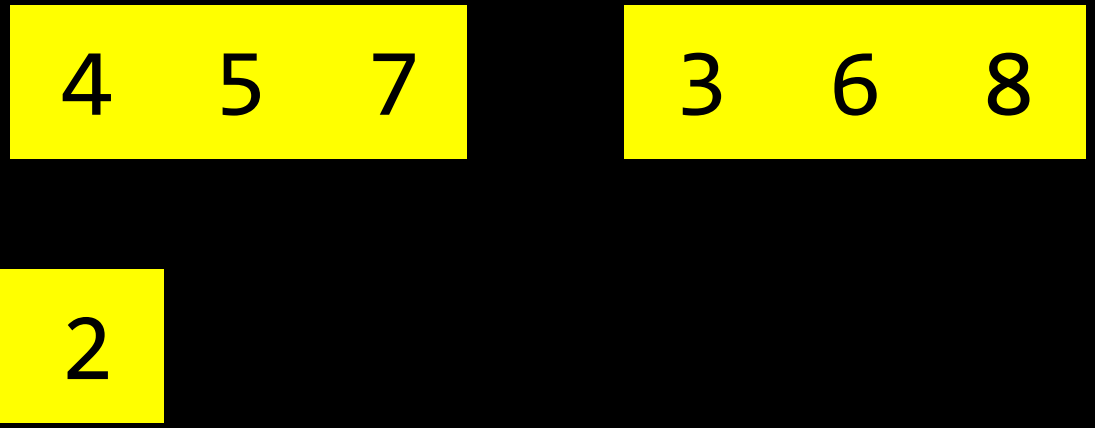

6
8
2
4
5
7
1
3

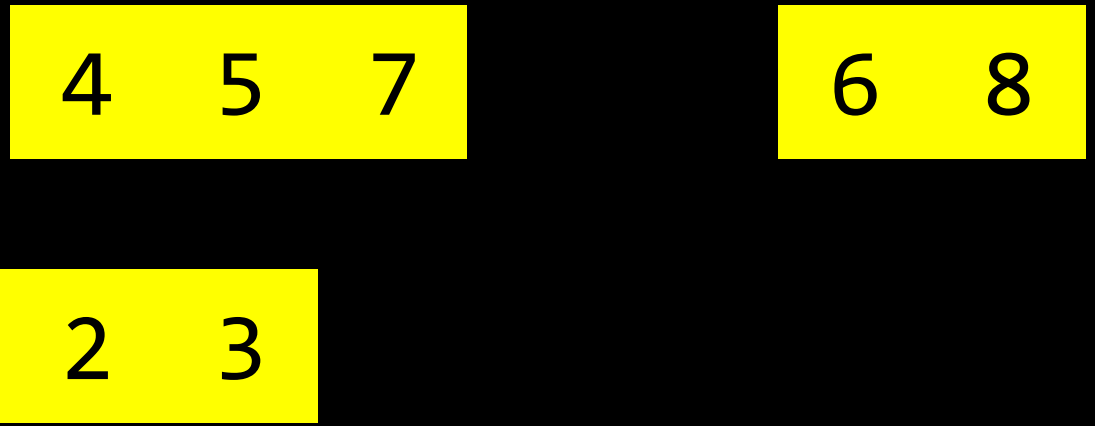

8
2 4 5 7 1 3 6

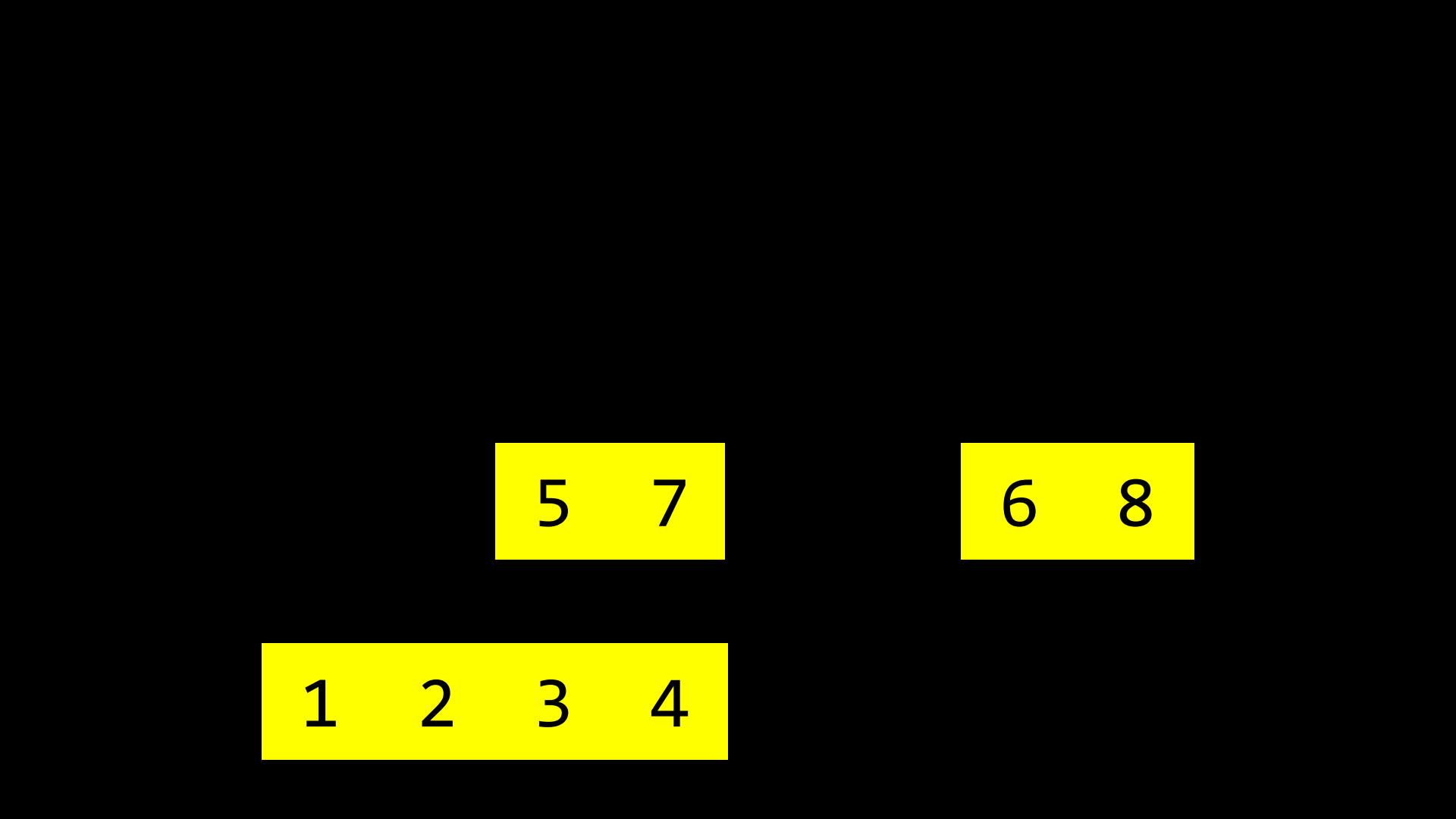

2 4 5 7 1 3 6 8

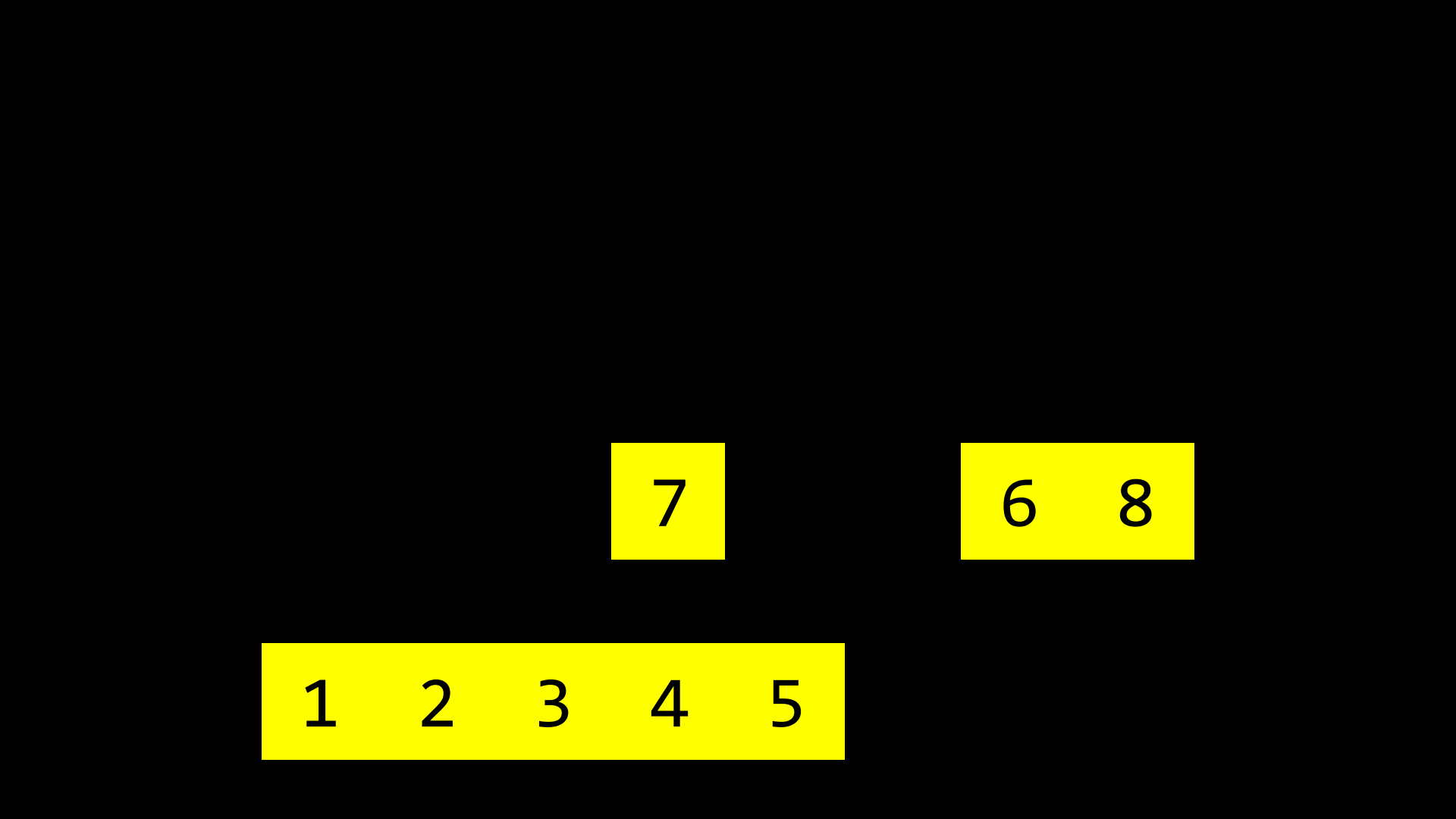

2 4 5 7 1 3 6 8

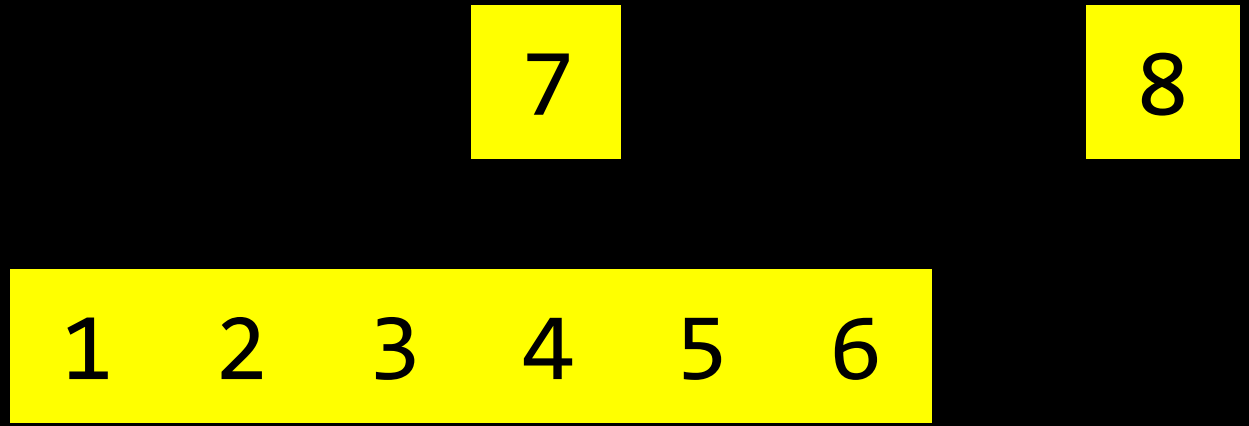

2 4 5 7
3 6 8
1


4 5 7
3 6 8
1 2


4 5 7
6 8
1 2 3

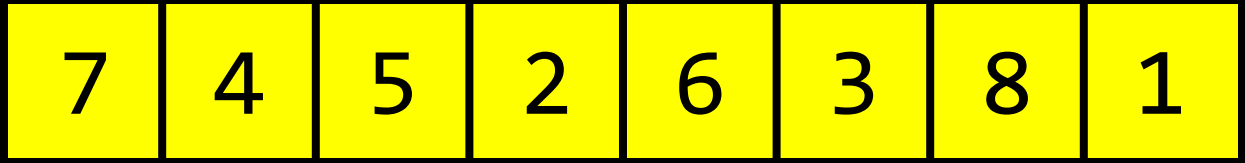

5 7
6 8
1 2 3 4

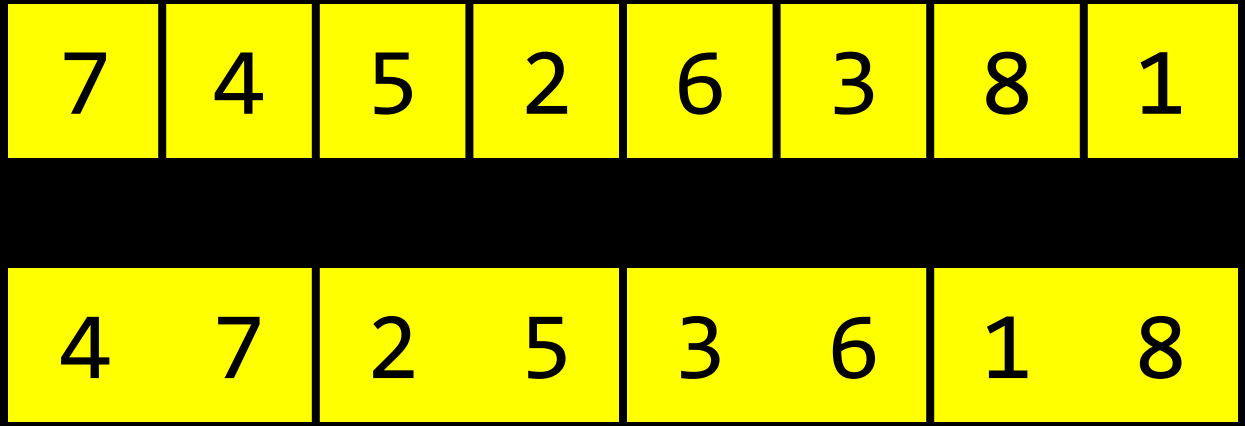

7
6 8
1 2 3 4 5

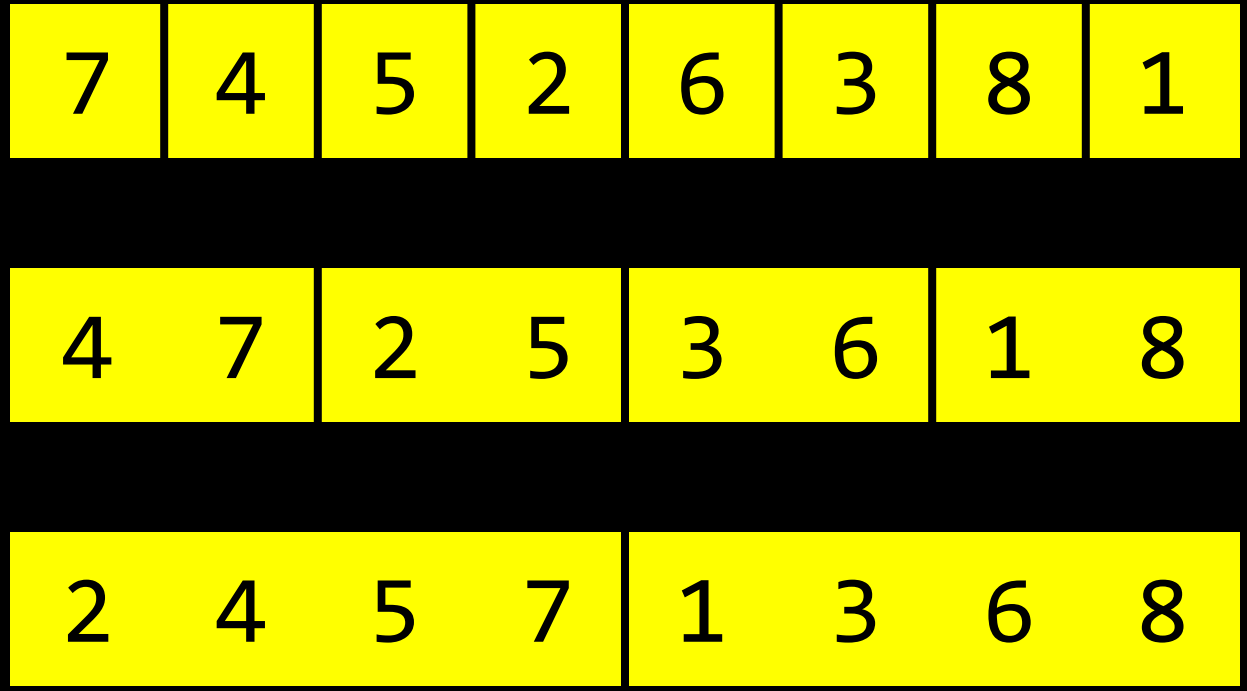

7
8
1 2 3 4 5 6

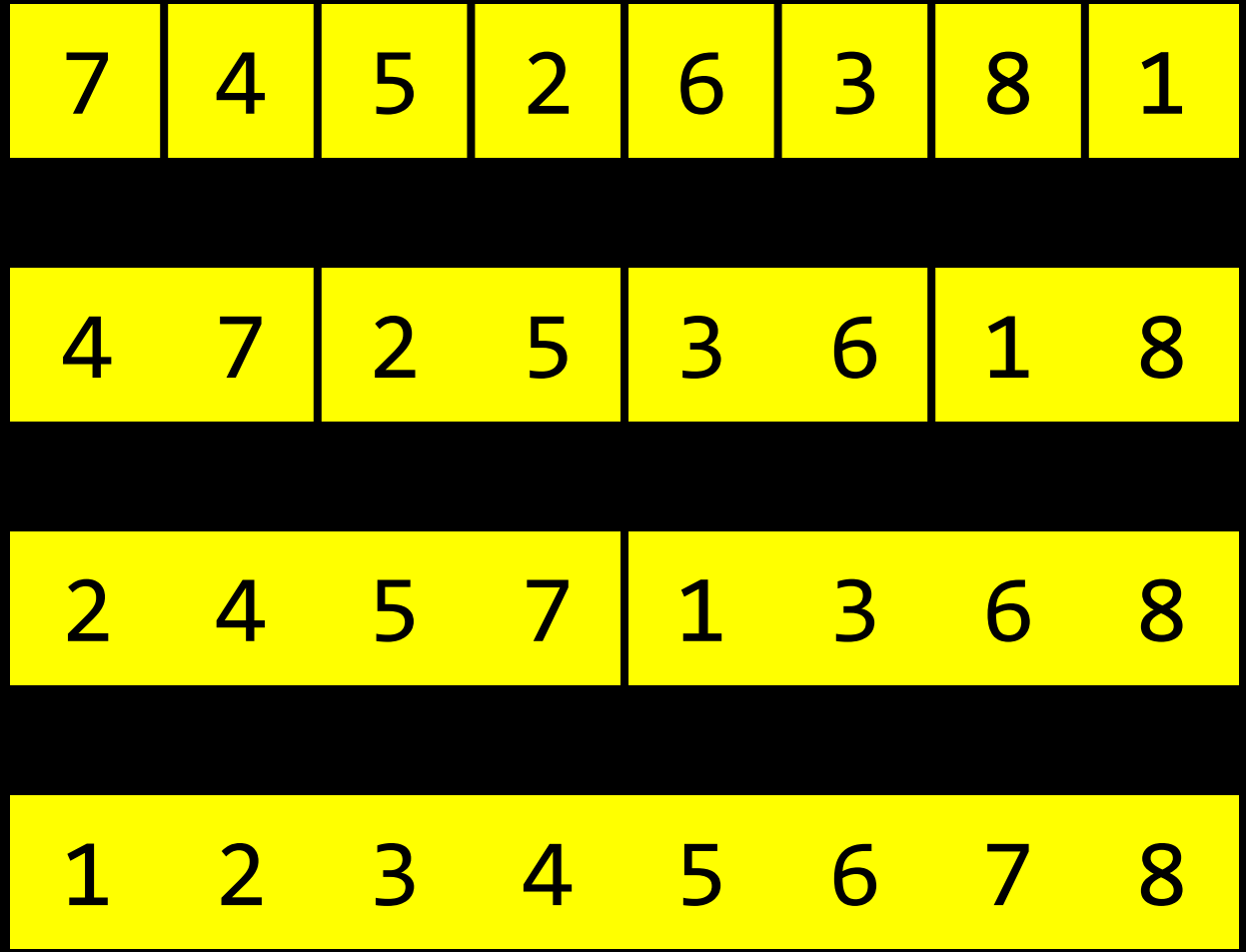

8
1 2 3 4 5 6 7

1 2 3 4 5 6 7 8


7
4
5
2
6
3
8
1


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


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


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

$O(n^2)$	bubble sort, selection sort
$O(n \log n)$	
$O(n)$	linear search
$O(\log n)$	binary search
$O(1)$	

$O(n^2)$	bubble sort, selection sort
$O(n \log n)$	merge sort
$O(n)$	linear search
$O(\log n)$	binary search
$O(1)$	

$\Omega(n^2)$	selection sort
$\Omega(n \log n)$	
$\Omega(n)$	bubble sort
$\Omega(\log n)$	
$\Omega(1)$	linear search, binary search

$\Omega(n^2)$	selection sort
$\Omega(n \log n)$	merge sort
$\Omega(n)$	bubble sort
$\Omega(\log n)$	
$\Omega(1)$	linear search, binary search

$\Theta(n^2)$

$\Theta(n \log n)$

$\Theta(n)$

$\Theta(\log n)$

$\Theta(1)$

$\Theta(n^2)$ selection sort

$\Theta(n \log n)$ merge sort

$\Theta(n)$

$\Theta(\log n)$

$\Theta(1)$