

Algorithm

Quicksort

Visual Walkthrough





Quicksort

- ♦ **In-place** comparison sorting algorithm.
- ♦ We pick a “pivot”, an element that we will use during the process to sort the list gradually.
- ♦ Reorder list so that all the elements with value less than the pivot come before the pivot and all the elements with value greater than the pivot come after the pivot. Equal elements can be in either side.
- ♦ Continue the process recursively for each portion of the list.





Quicksort

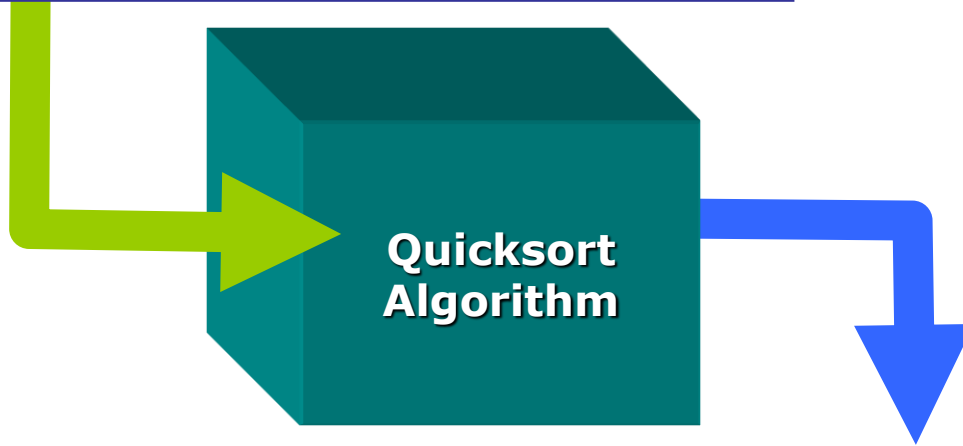
- ♦ Efficiency depends on the pivot that you choose.
- ♦ Pivot can be chosen arbitrarily, but you can make the algorithm more efficient by choosing the pivot strategically.
- ♦ Efficient to sort large lists if the right pivot is chosen.





Quicksort

[7, 2, 8, 1, 0, 3, 5]

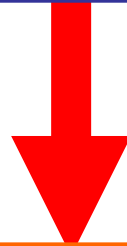


[0, 1, 2, 3, 5, 7, 8]



Quicksort

[7, 2, 8, 1, 0, 3, 5]



Sort

[0, 1, 2, 3, 5, 7, 8]



Quicksort

```
[7, 2, 8, 1, 0, 3, 5]
```



Quicksort

```
[7, 2, 8, 1, 0, 3, 5]
```

[0]

[1]

[2]

[3]

[4]

[5]

[6]



Quicksort

[7, 2, 8, 1, 0, 3, 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]



Quicksort

[7, 2, 8, 1, 0, 3, 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]

$i = -1$

$j = 0$



Quicksort

[7, 2, 8, 1, 0, 3, 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]



$i = -1$

$j = 0$



Quicksort

$7 \leq 5$? No!

[7, 2, 8, 1, 0, 3, 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]



$i = -1$

$j = 0$



Quicksort

[7, 2, 8, 1, 0, 3, 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]



$i = -1$

$j = 1$



Quicksort

$2 \leq 5$? Yes!

[7, 2, 8, 1, 0, 3, 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]



$i = -1$

$j = 1$



Quicksort

[7, 2, 8, 1, 0, 3, 5]

[0]



[1]



[2]

[3]

[4]

[5]

[6]

$i = 0$

$j = 1$



Quicksort



[0]

[1]

[2]

[3]

[4]

[5]

[6]



$i = 0$

$j = 1$



Quicksort

[2 , 7 , 8 , 1 , 0 , 3 , 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]



$i = 0$

$j = 2$



Quicksort

$8 \leq 5$? No!

[2 , 7 , 8 , 1 , 0 , 3 , 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]



$i = 0$

$j = 2$



Quicksort

[2 , 7 , 8 , 1 , 0 , 3 , 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]



$i = 0$

$j = 3$



Quicksort

1 ≤ 5? Yes!

[2, 7, 8, 1, 0, 3, 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]



i = 0

j = 3



Quicksort

$1 \leq 5$? Yes!

[2, 7, 8, 1, 0, 3, 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]

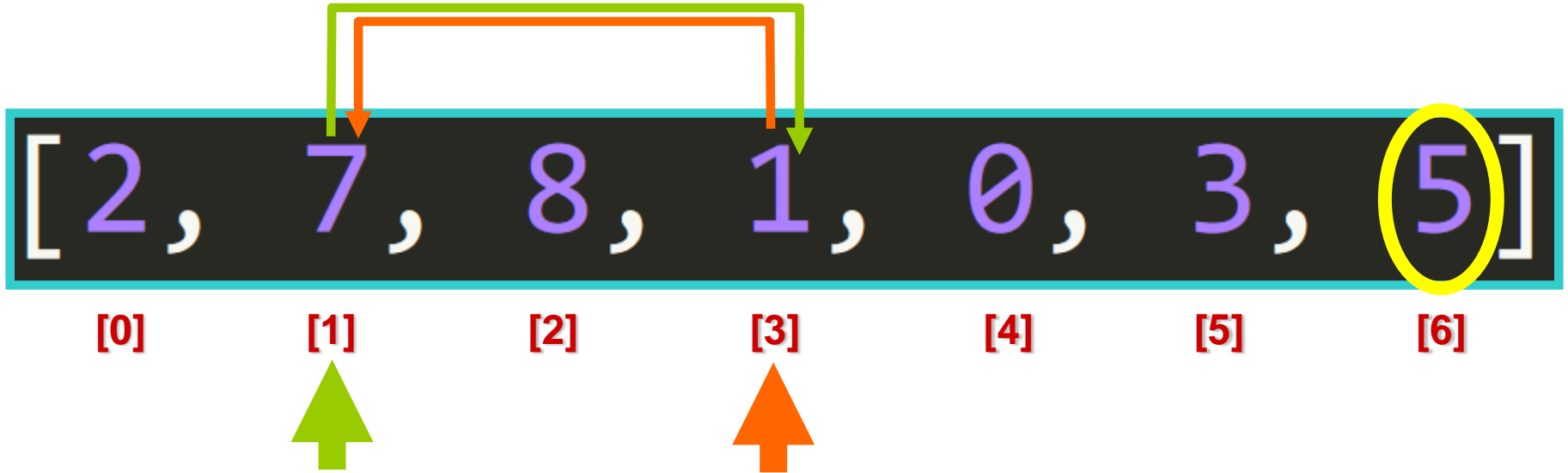


$i = 1$

$j = 3$



Quicksort



[0]

[1]

[2]

[3]

[4]

[5]

[6]

$i = 1$

$j = 3$



Quicksort

[2, 1, 8, 7, 0, 3, 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]



$i = 1$

$j = 3$



Quicksort

[2, 1, 8, 7, 0, 3, 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]



$i = 1$

$j = 4$



Quicksort

[2, 1, 8, 7, 0, 3, 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]



0 ≤ 5? Yes!

i = 1

j = 4



Quicksort

[2, 1, 8, 7, 0, 3, 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]



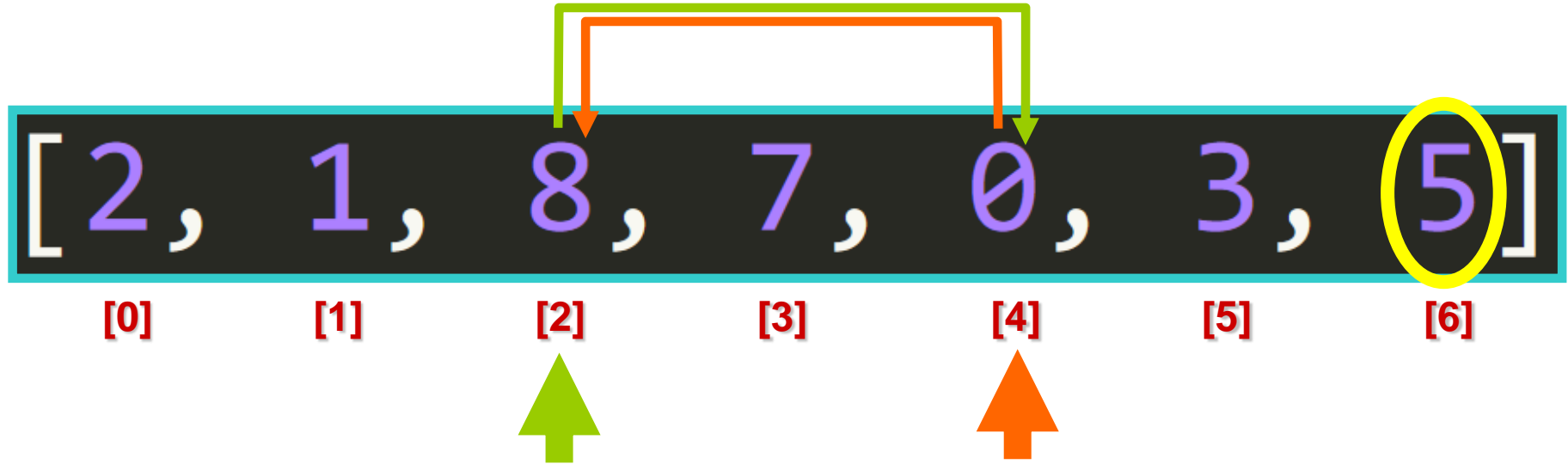
0 <= 5? Yes!

i = 2

j = 4



Quicksort



$i = 2$

$j = 4$



Quicksort

[2, 1, 0, 7, 8, 3, 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]



$i = 2$

$j = 4$



Quicksort

[2, 1, 0, 7, 8, 3, 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]



$i = 2$

$j = 5$



Quicksort

3 ≤ 5? Yes!

[2, 1, 0, 7, 8, 3, 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]



i = 2

j = 5



Quicksort

3 ≤ 5? Yes!

[2, 1, 0, 7, 8, 3, 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]

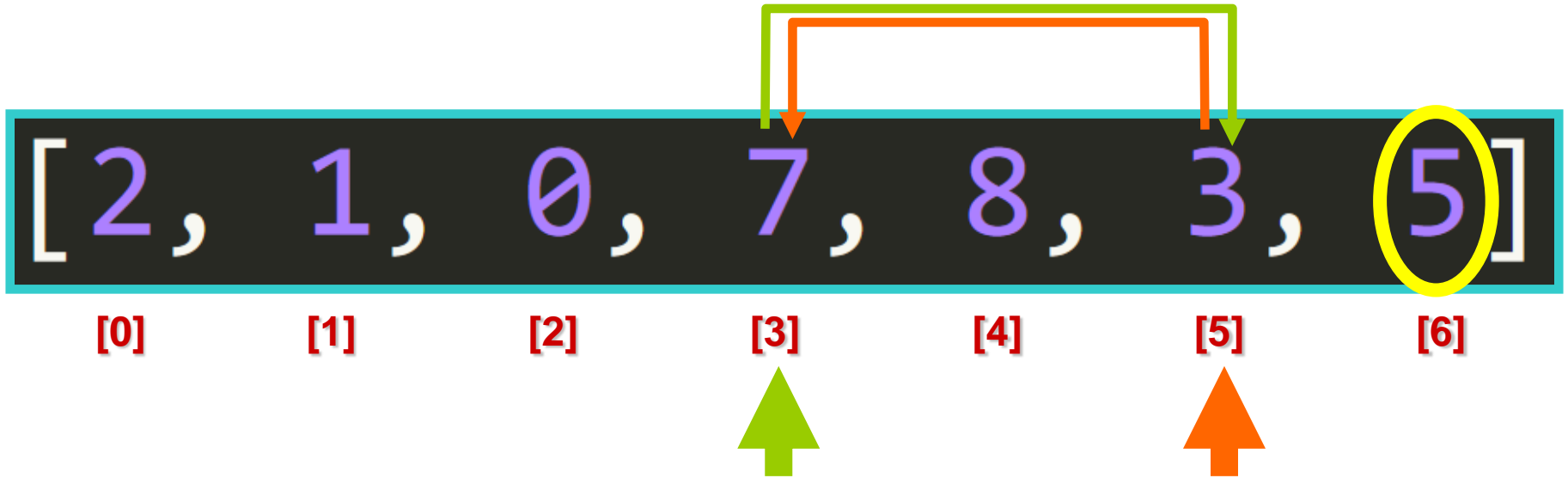


i = 3

j = 5



Quicksort



$i = 3$

$j = 5$



Quicksort

[2, 1, 0, 3, 8, 7, 5]

[0]

[1]

[2]

[3]

[4]

[5]

[6]

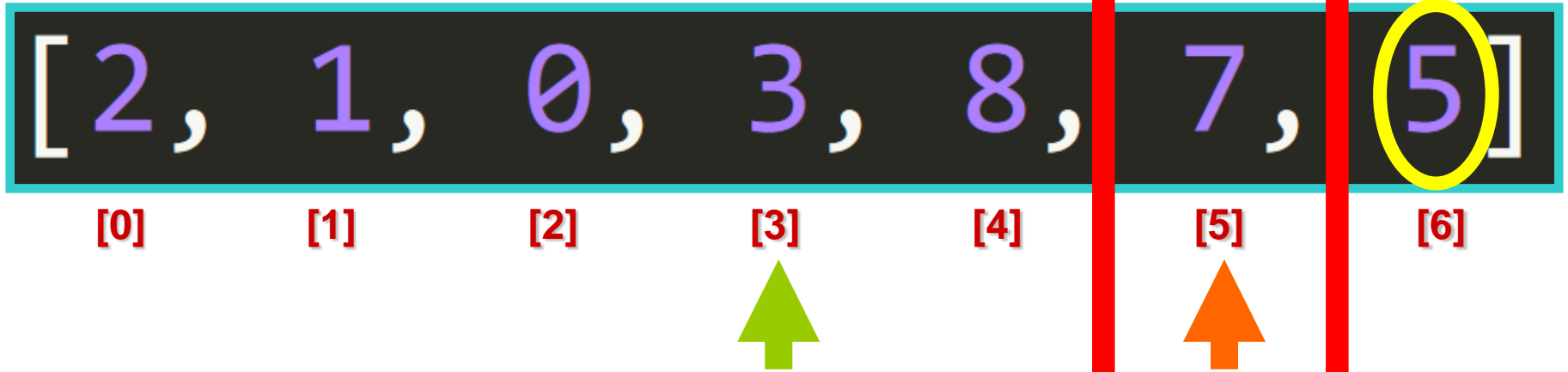


$i = 3$

$j = 5$



Quicksort

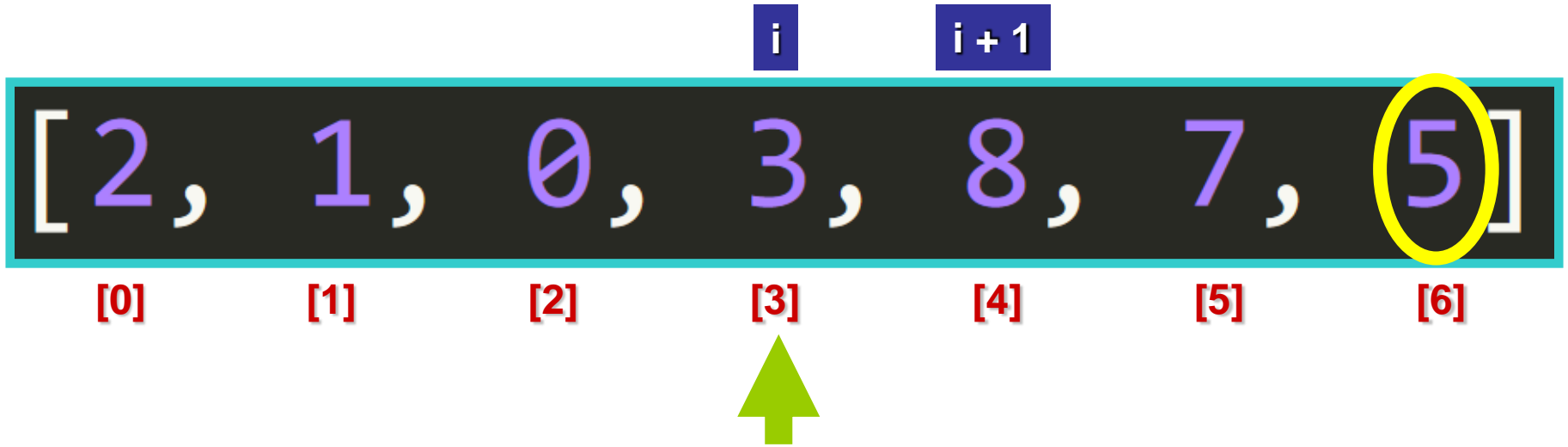


`i = 3`

`j = 5`



Quicksort



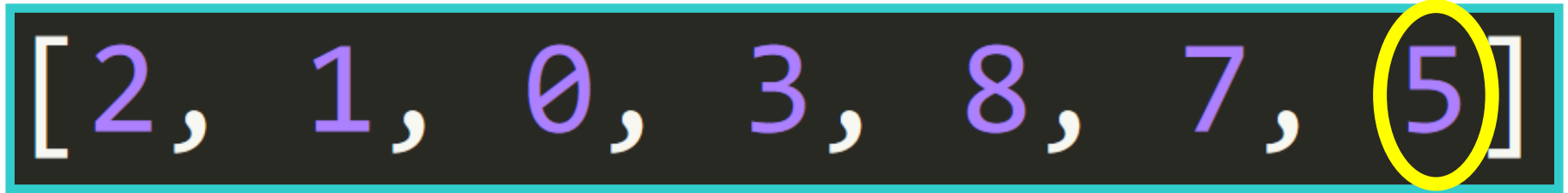
$i = 3$

$j = 5$



Quicksort

$i + 1$



[0]

[1]

[2]

[3]

[4]

[5]

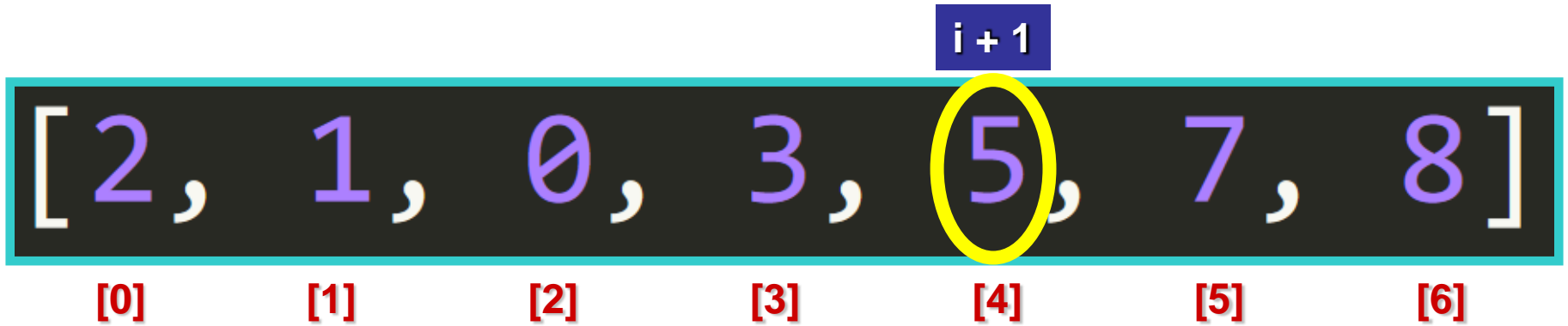
[6]

$i = 3$

$j = 5$



Quicksort



$i = 3$

$j = 5$



Quicksort

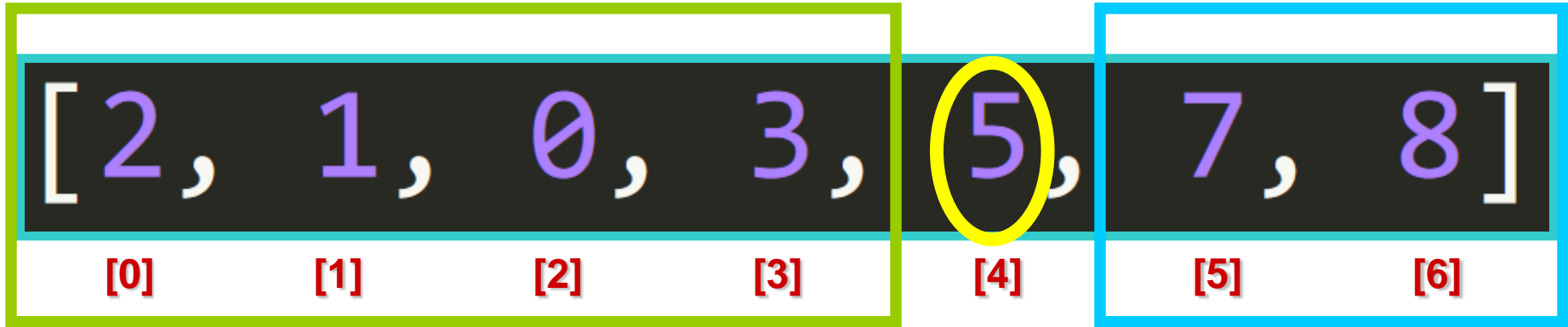


$i = 3$

$j = 5$



Quicksort





Quicksort

[2, 1, 0, 3, 5, 7, 8]

[0]

[1]

[2]

[3]

[4]

[5]

[6]



Quicksort

[2, 1, 0, 3, 5, 7, 8]

[0]

[1]

[2]

[3]

[4]

[5]

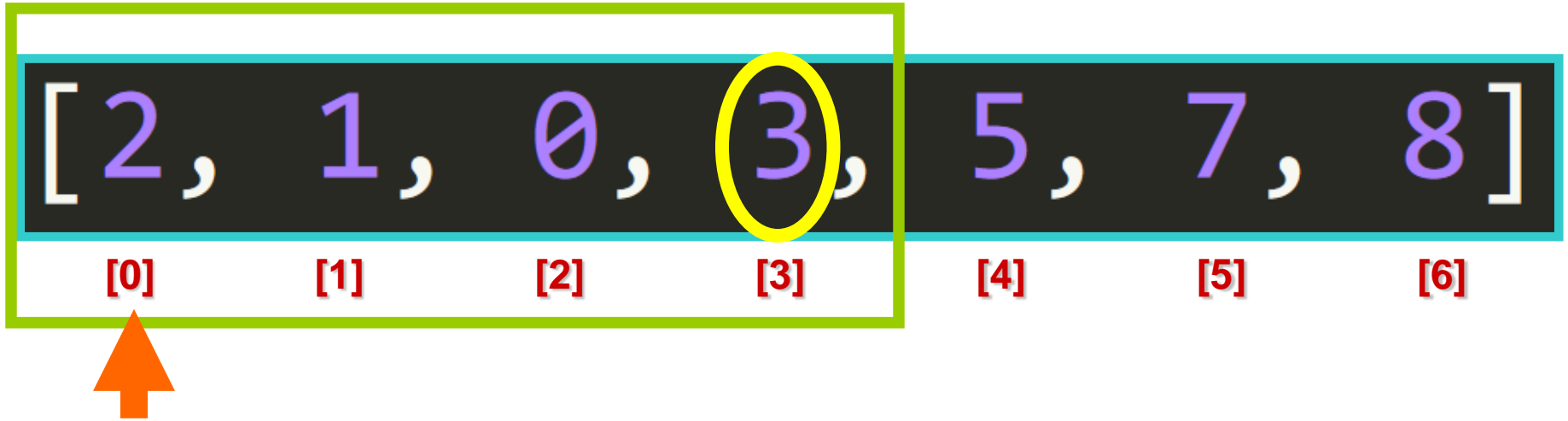
[6]

$i = -1$

$j = 0$



Quicksort



$i = -1$

$j = 0$



Quicksort

$2 \leq 3$? Yes!

[2, 1, 0, 3, 5, 7, 8]

[0]

[1]

[2]

[3]

[4]

[5]

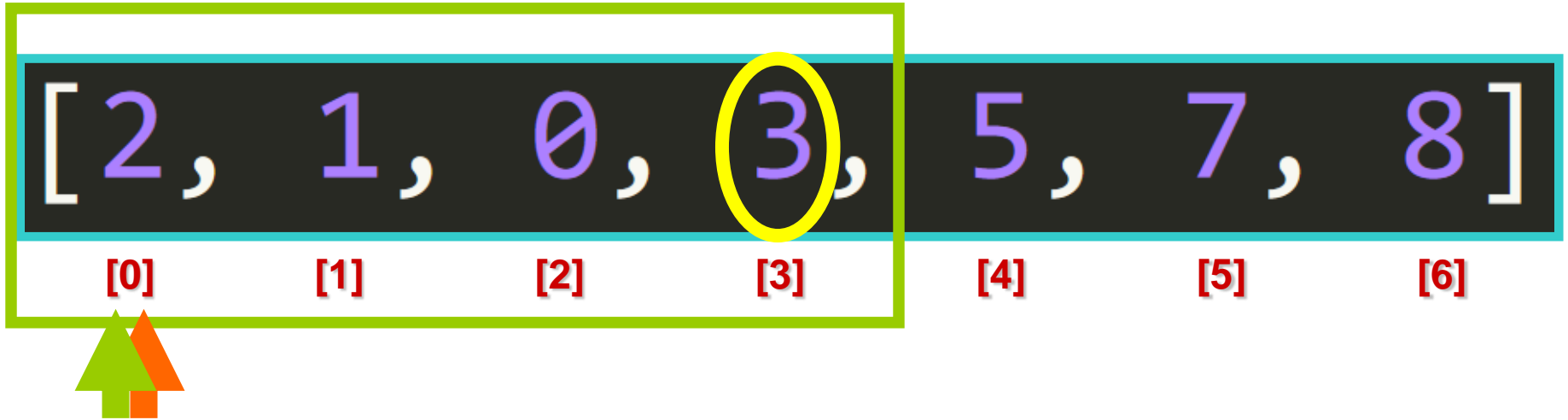
[6]

$i = -1$

$j = 0$



Quicksort

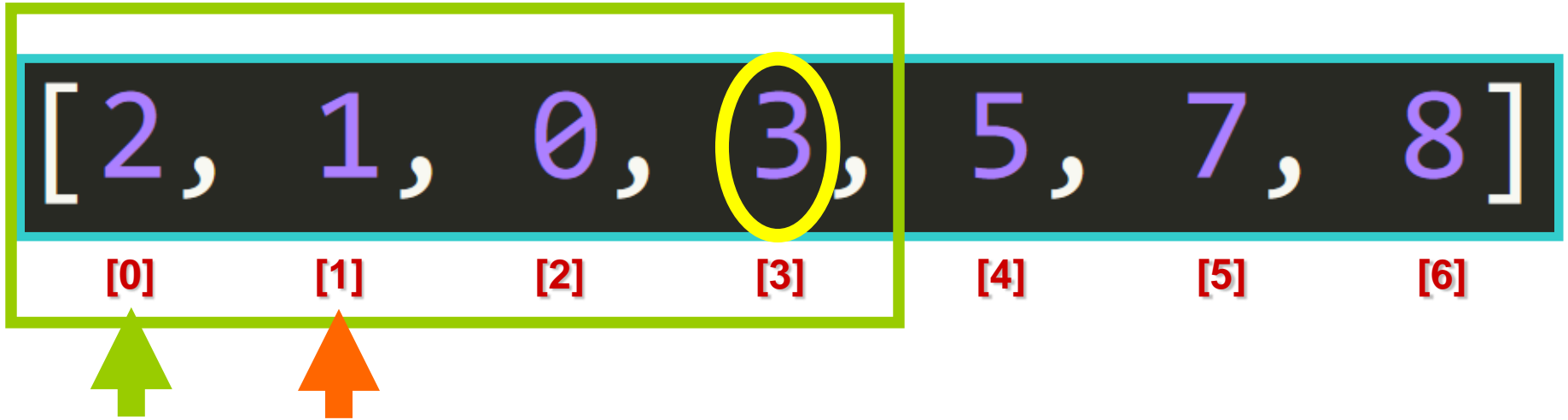


`i = 0`

`j = 0`



Quicksort



`i = 0`

`j = 1`



Quicksort

1 ≤ 3? Yes!

[2, 1, 0, 3, 5, 7, 8]

[0]

[1]

[2]

[3]

[4]

[5]

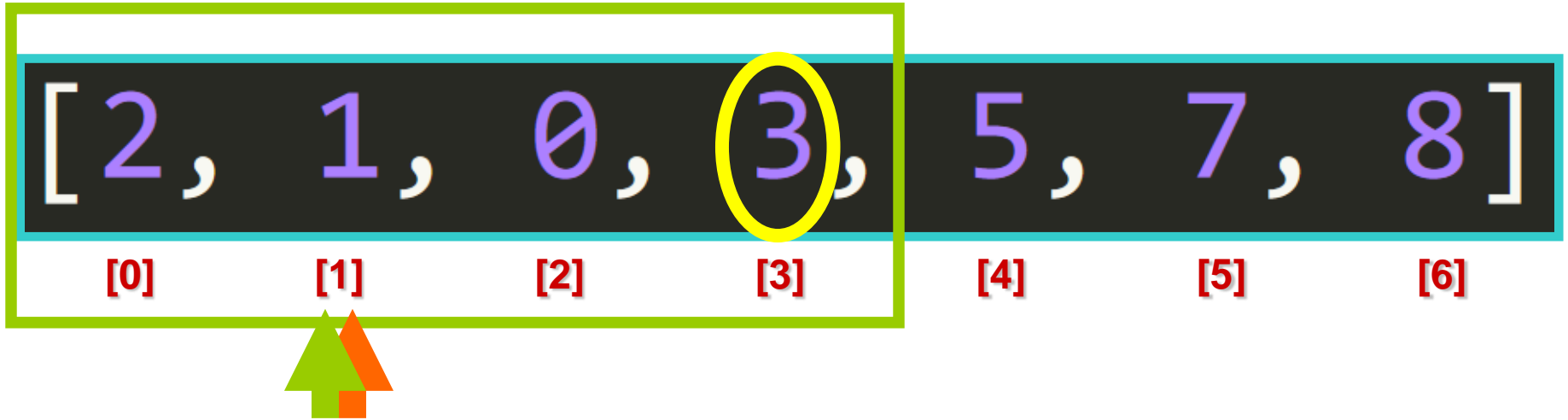
[6]

i = 0

j = 1



Quicksort



`i = 1`

`j = 1`



Quicksort

$0 \leq 3$? Yes!

[2 , 1 , 0 , 3 , 5 , 7 , 8]

[0]

[1]

[2]

[3]

[4]

[5]

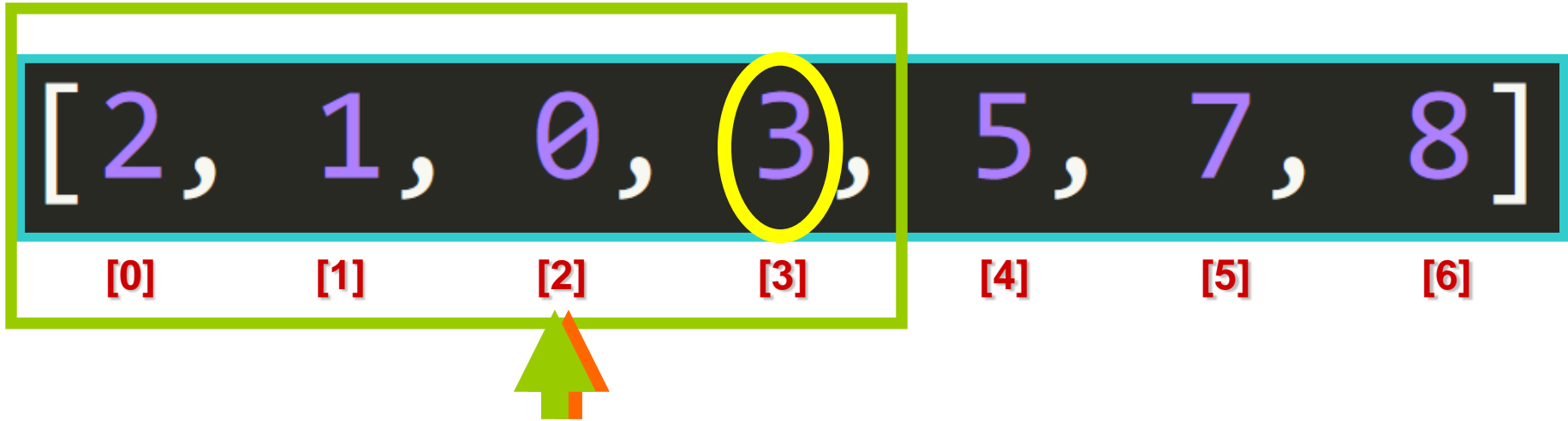
[6]

$i = 1$

$j = 2$



Quicksort



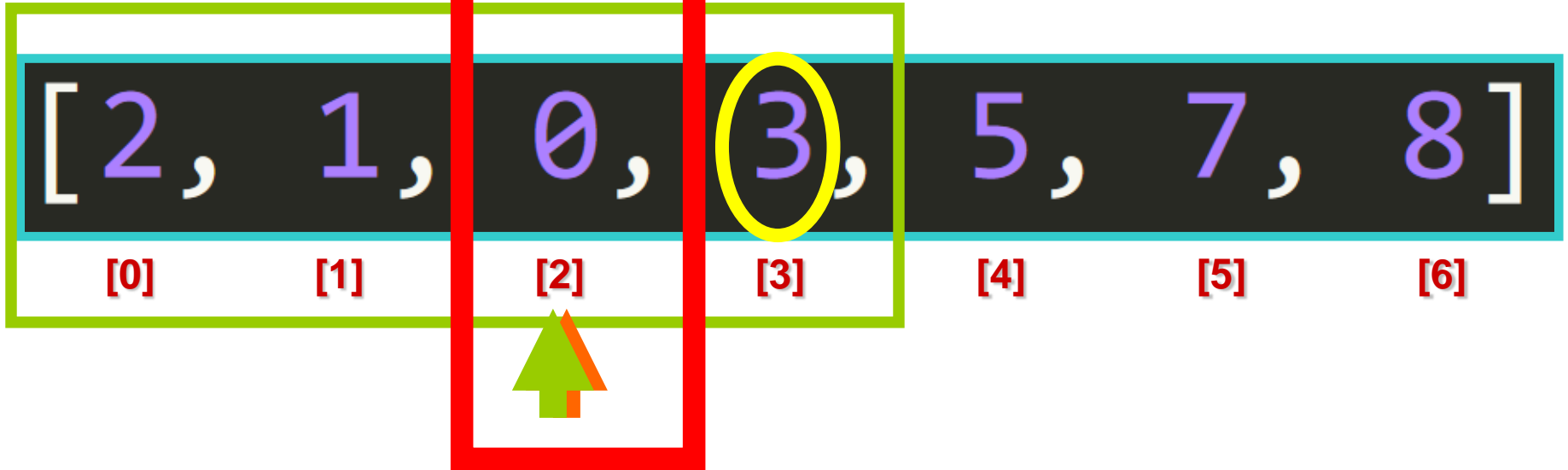
$i = 2$

$j = 2$



Quicksort

Stop!

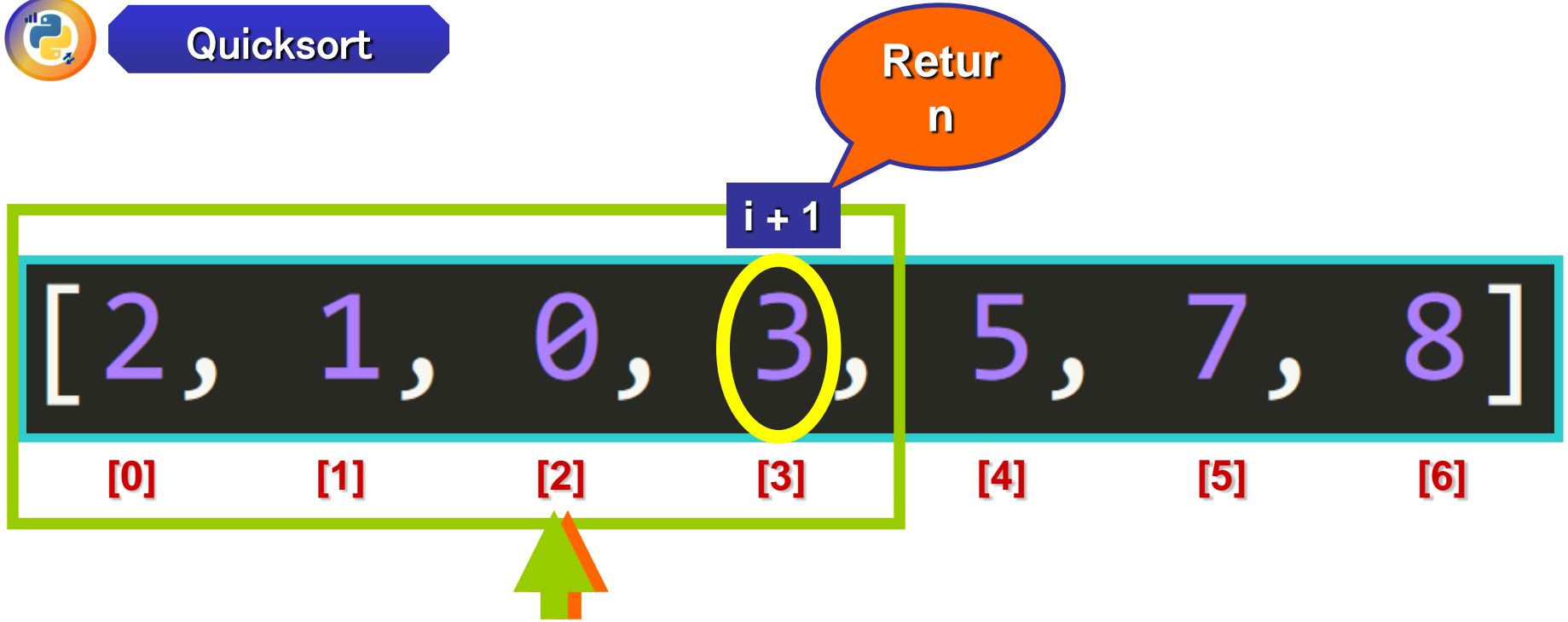


$i = 2$

$j = 2$



Quicksort



`i = 2`

`j = 2`



Quicksort

[2, 1, 0, 3, 5, 7, 8]

[0]

[1]

[2]

[3]

[4]

[5]

[6]

$i = -1$

$j = 0$



Quicksort

[2, 1, 0, 3, 5, 7, 8]

[0]

[1]

[2]

[3]

[4]

[5]

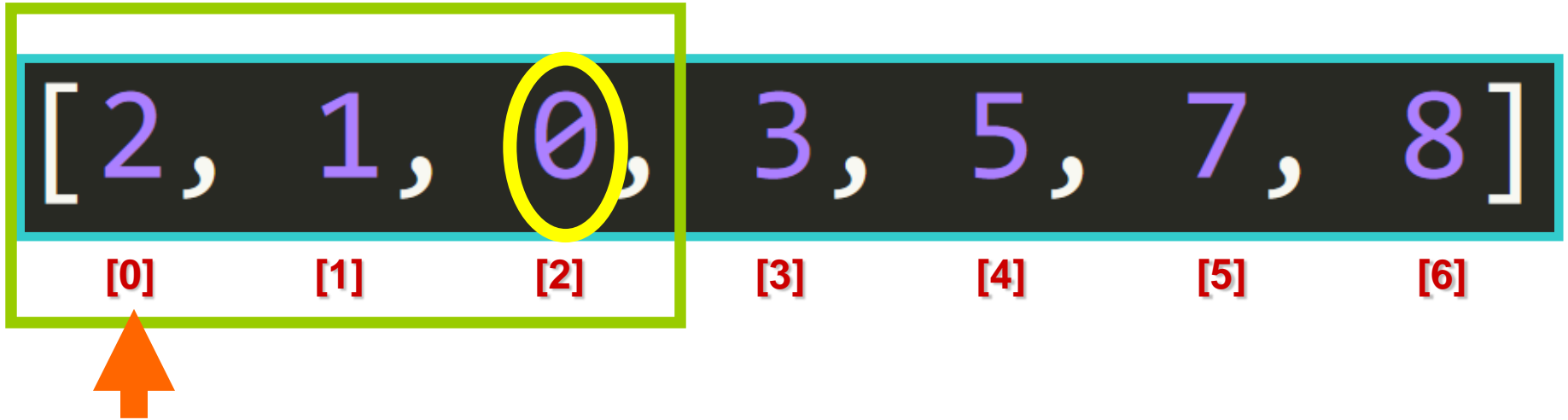
[6]

$i = -1$

$j = 0$



Quicksort



`i = -1`

`j = 0`



Quicksort

$2 \leq 0$? No!

[2 , 1 , 0 , 3 , 5 , 7 , 8]

[0]

[1]

[2]

[3]

[4]

[5]

[6]

$i = -1$

$j = 0$



Quicksort

$2 \leq 0$? No!

[2 , 1 , 0 , 3 , 5 , 7 , 8]

[0]

[1]

[2]

[3]

[4]

[5]

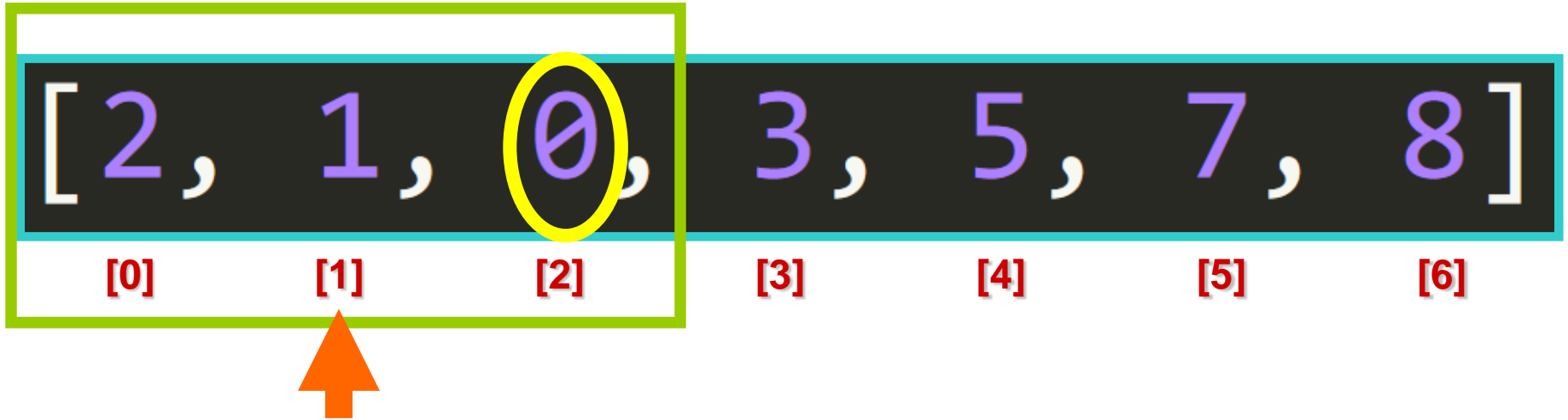
[6]

$i = -1$

$j = 0$



Quicksort



`i = -1`

`j = 1`



Quicksort

$1 \leq 0$? No!

[2, 1, 0, 3, 5, 7, 8]

[0]

[1]

[2]

[3]

[4]

[5]

[6]

$i = -1$

$j = 1$

Stop!

[2, 1, 0, 3, 5, 7, 8]

[0]

[1]

[2]

[3]

[4]

[5]

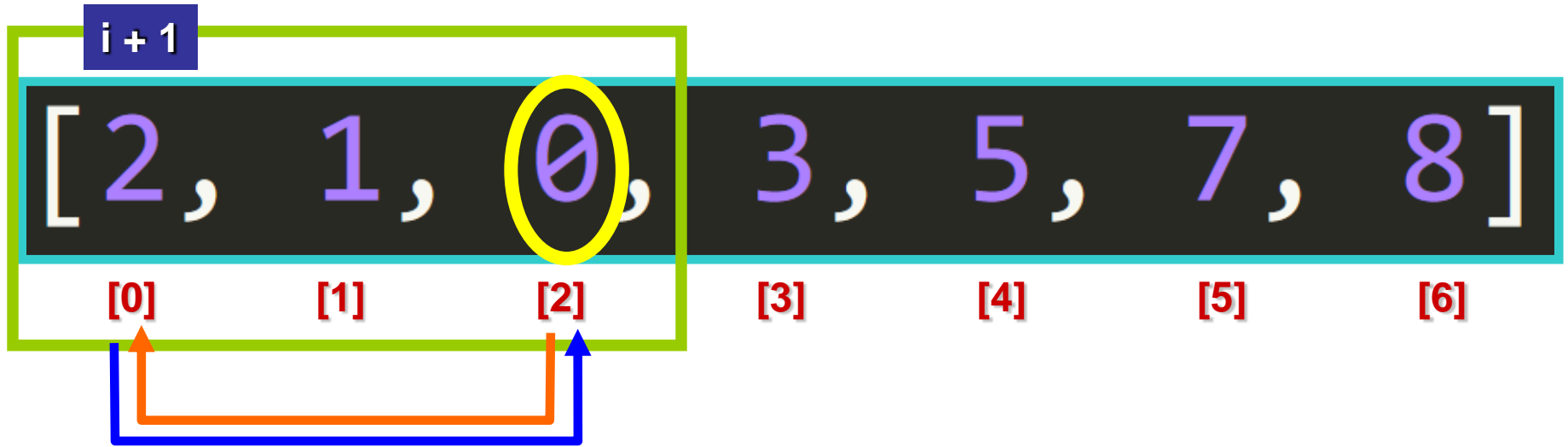
[6]

$i = -1$

$j = 1$



Quicksort

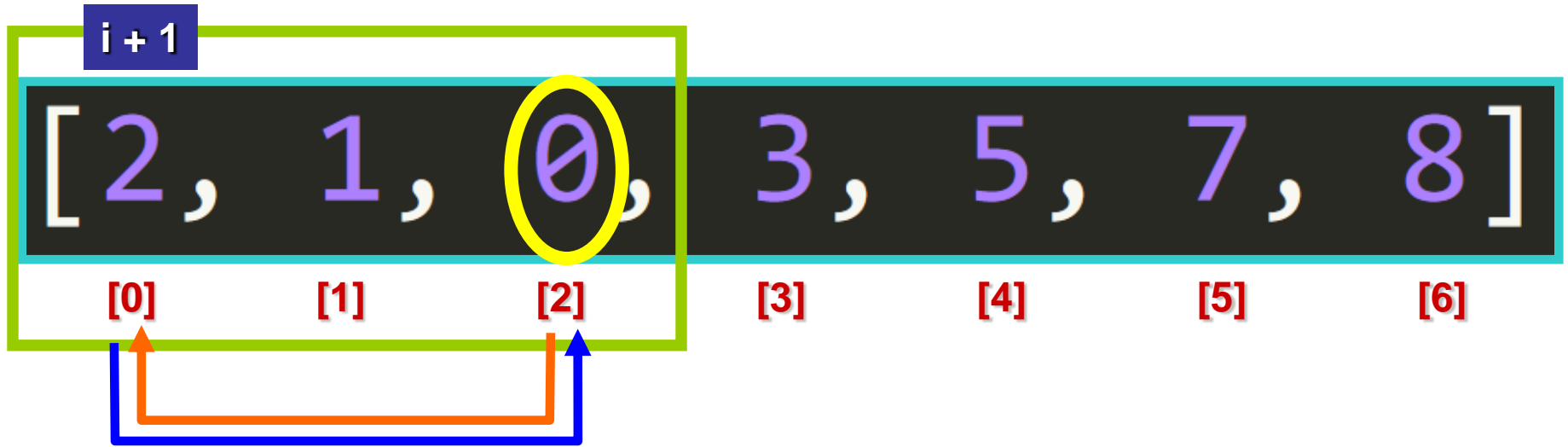


`i = -1`

`j = 1`



Quicksort

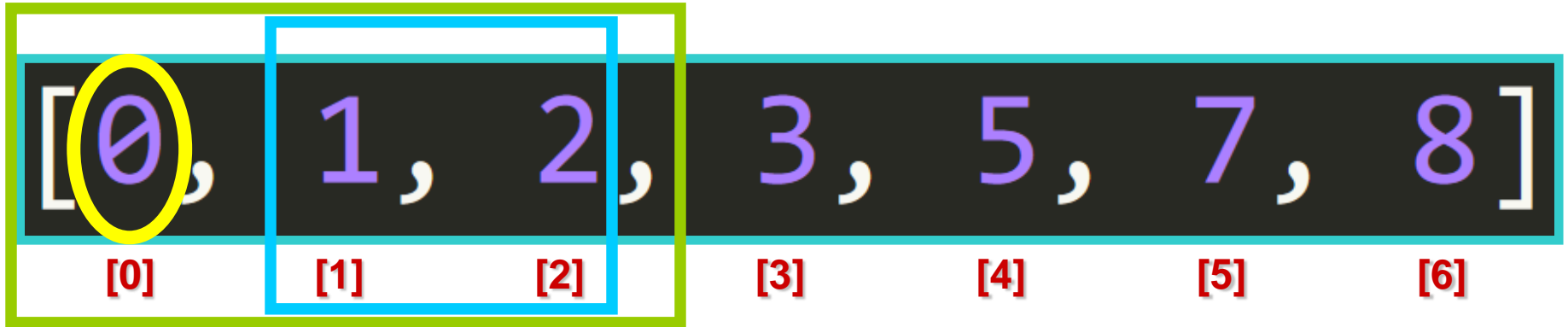


`i = -1`

`j = 1`



Quicksort

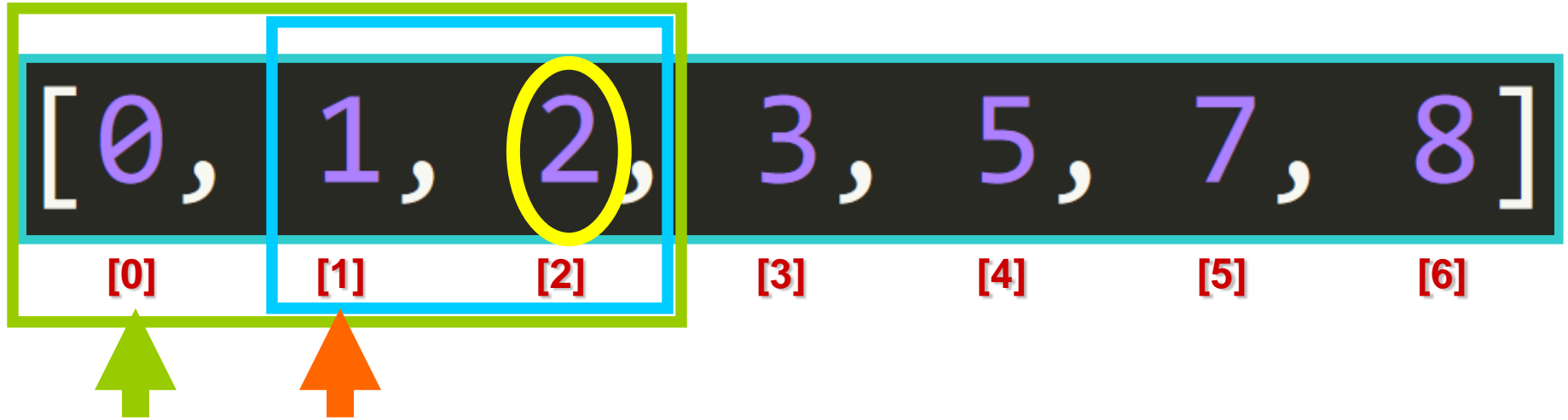


$i = -1$

$j = 1$



Quicksort



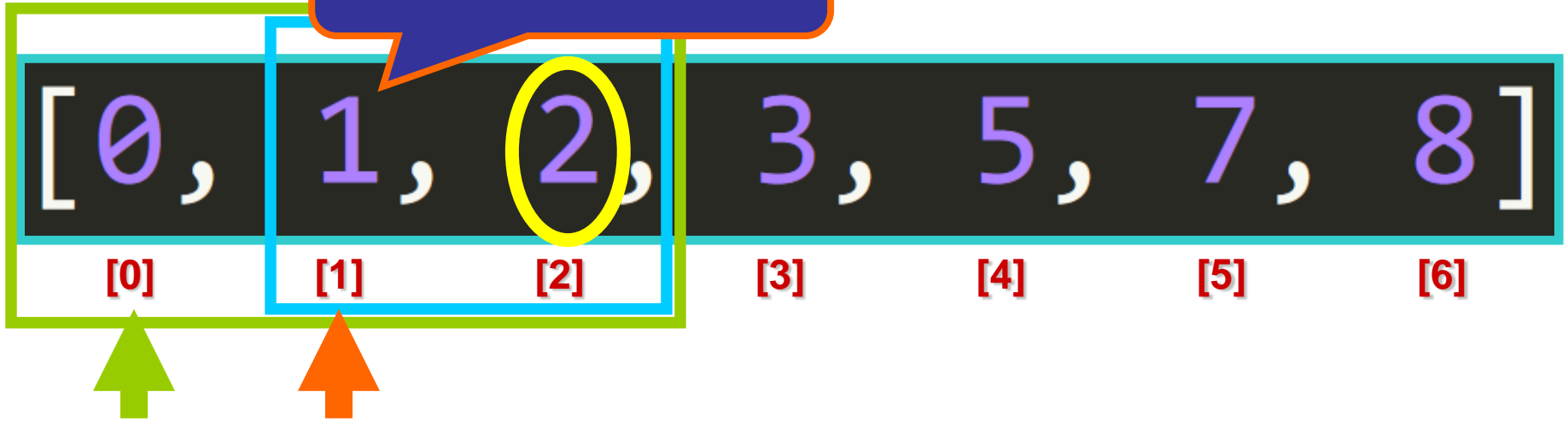
`i = 0`

`j = 1`



Quicksort

1 <= 2? Yes!

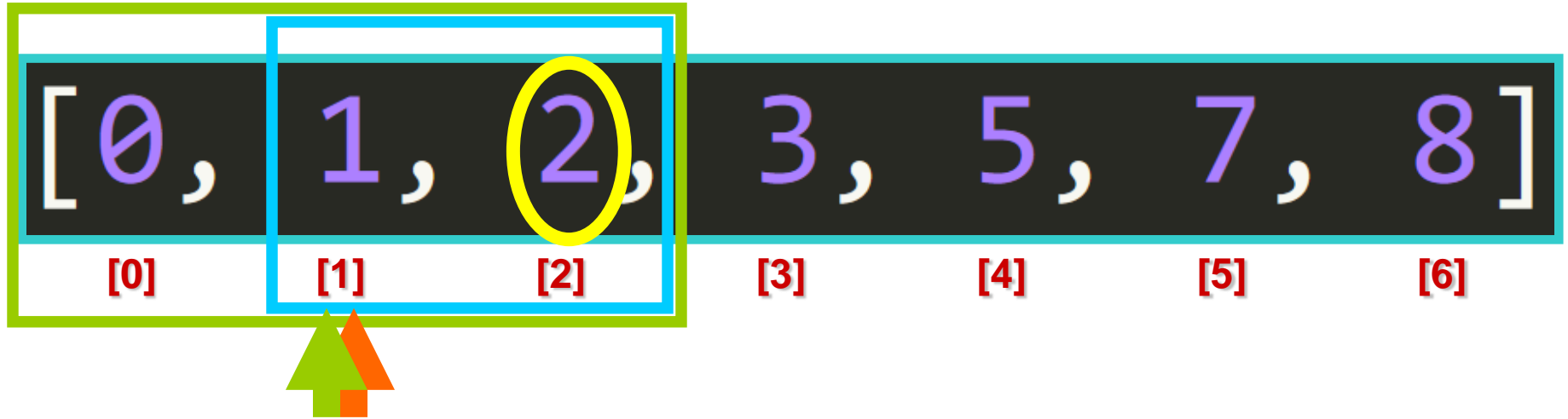


`i = 0`

`j = 1`



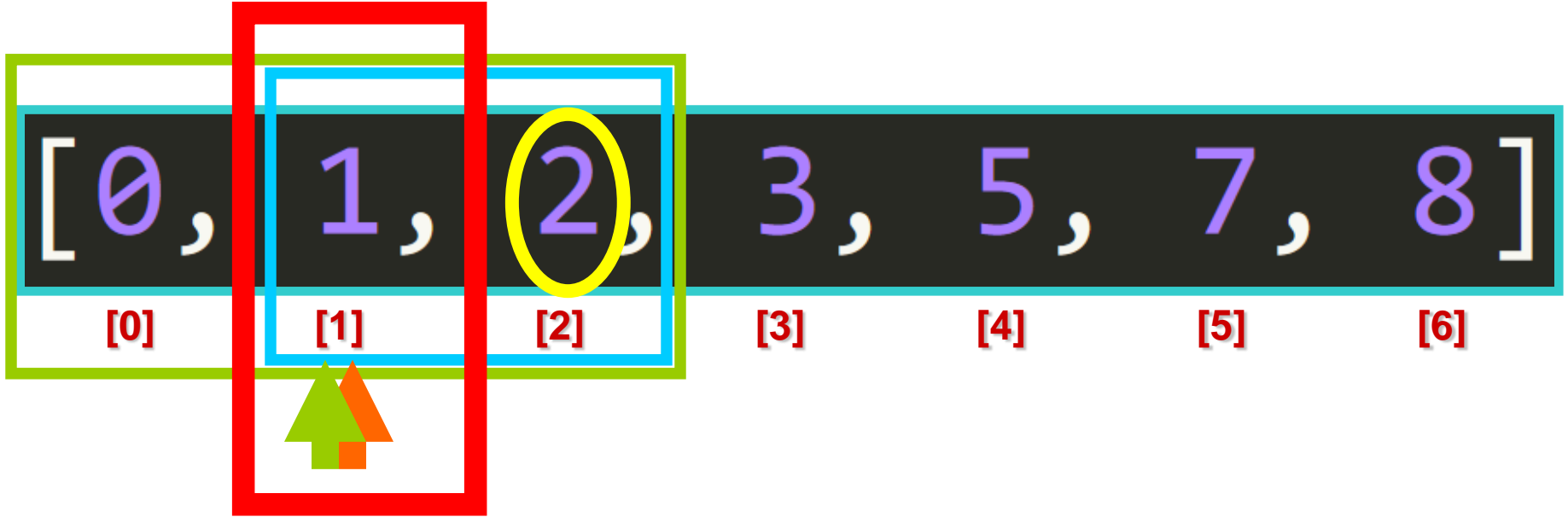
Quicksort



`i = 1`

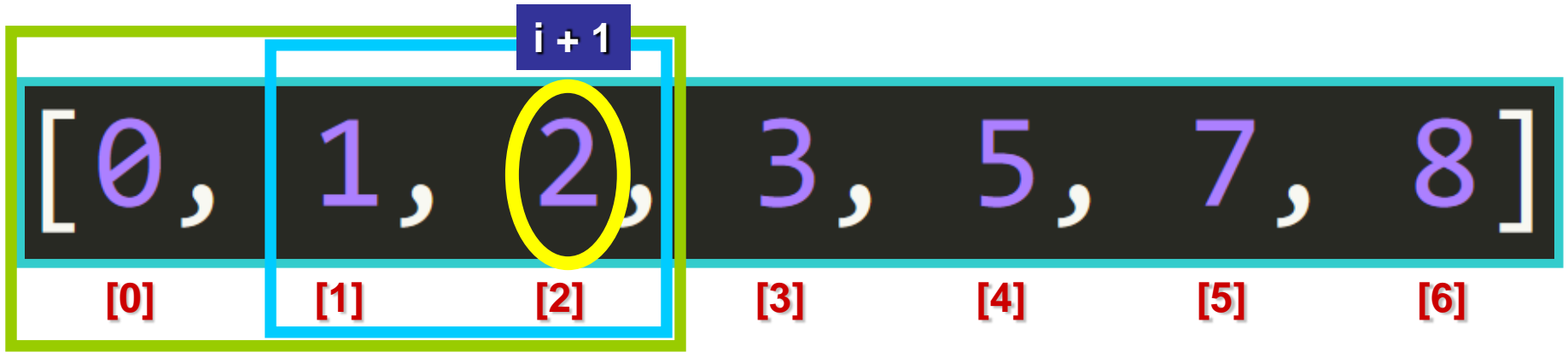
`j = 1`

Stop!



$i = 1$

$j = 1$



$i = 1$

$j = 1$



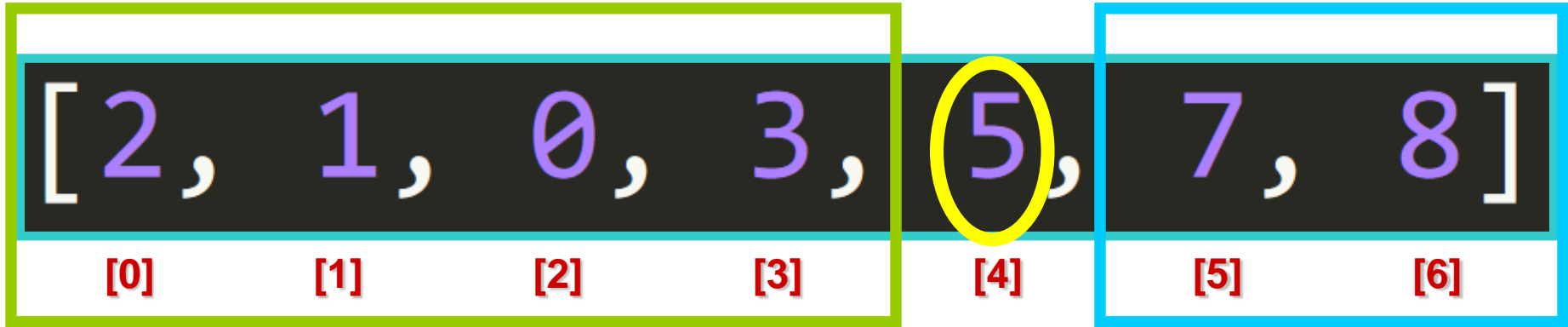
Quicksort

A few steps ago...



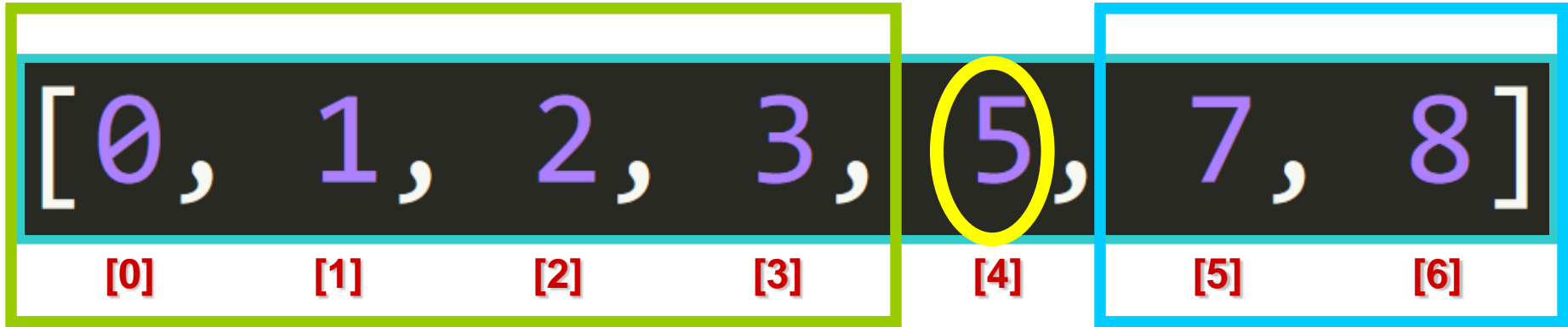


Quicksort



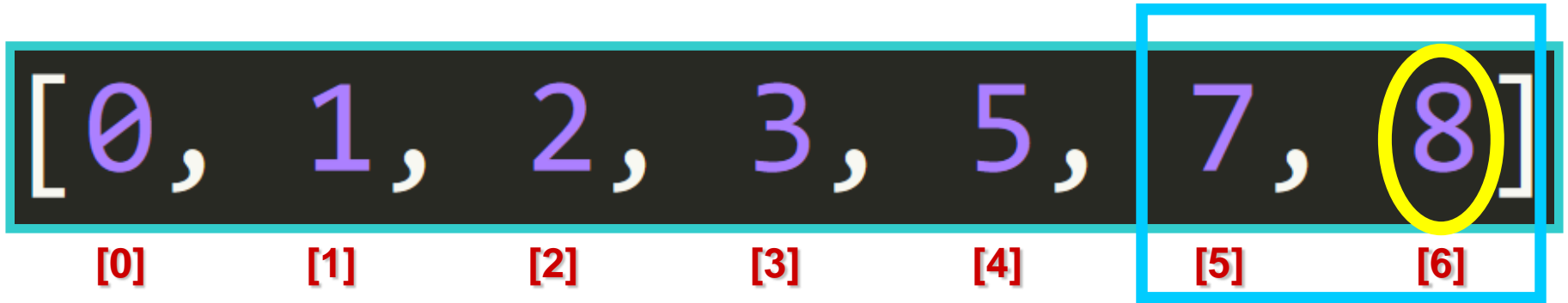


Quicksort





Quicksort

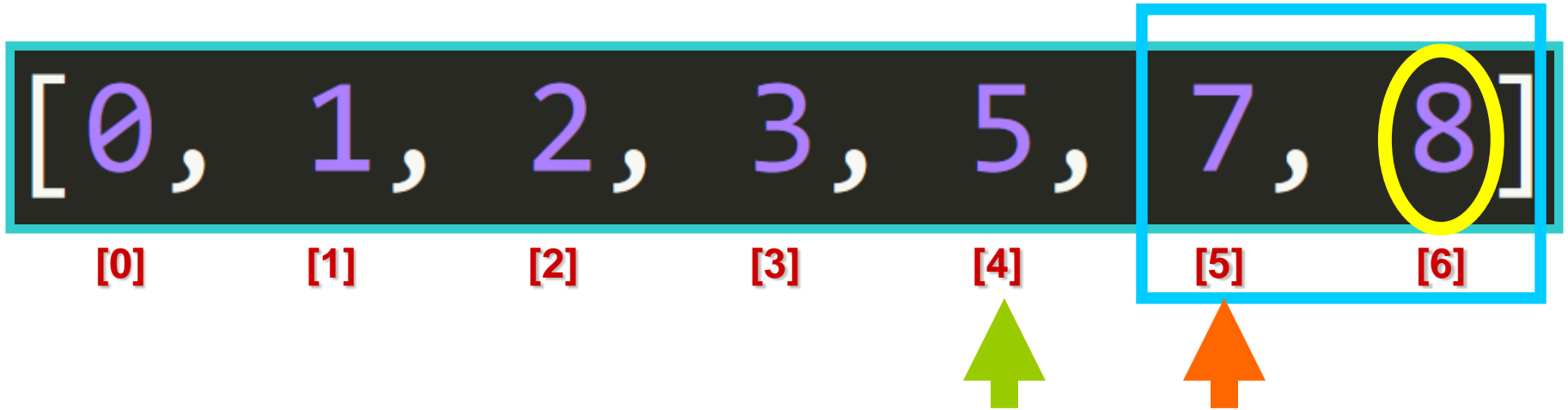


$i = 4$

$j = 5$



Quicksort

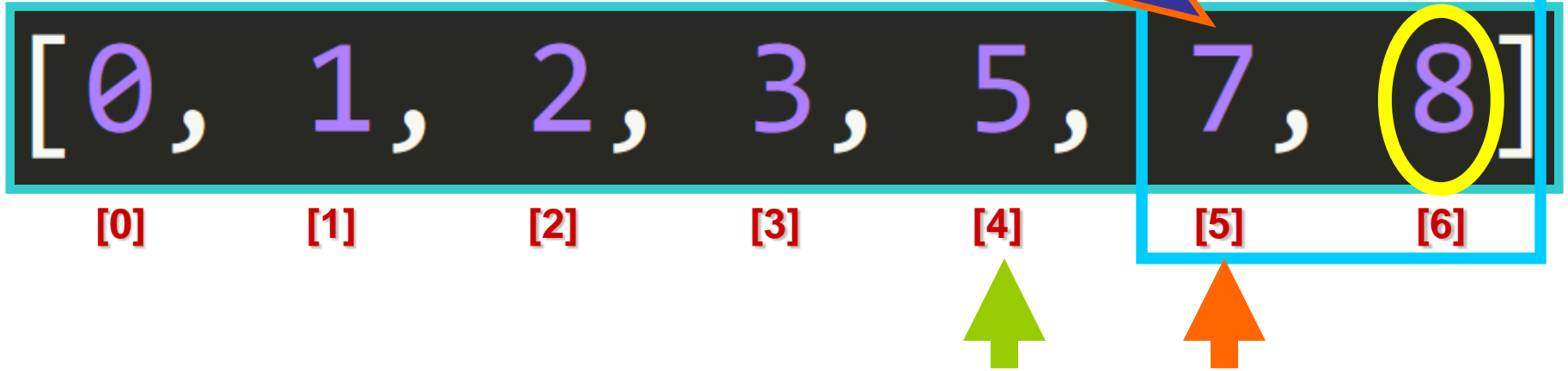


$i = 4$

$j = 5$



Quicksort

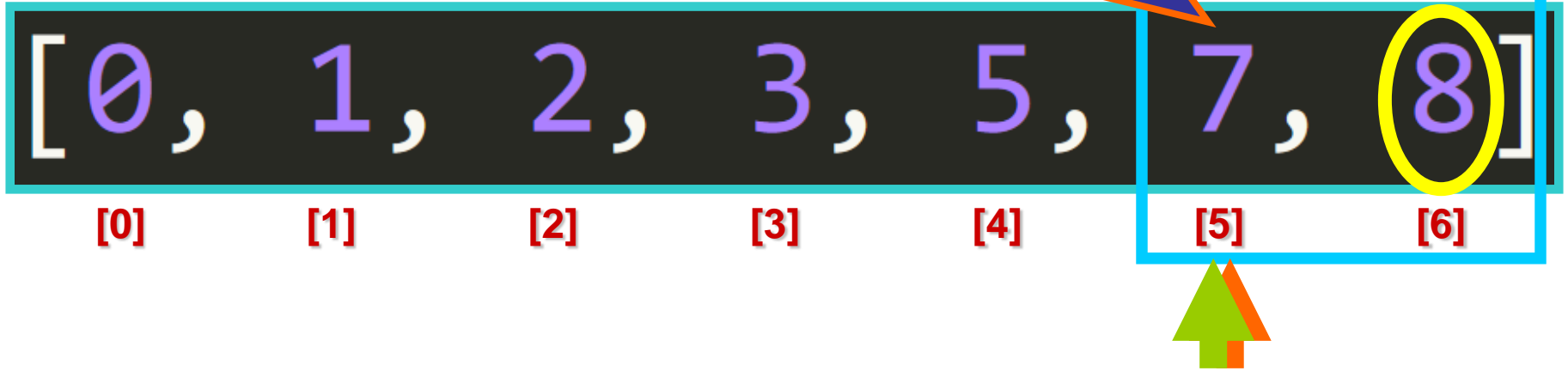


$i = 4$

$j = 5$



Quicksort



`i = 5`

`j = 5`



Quicksort

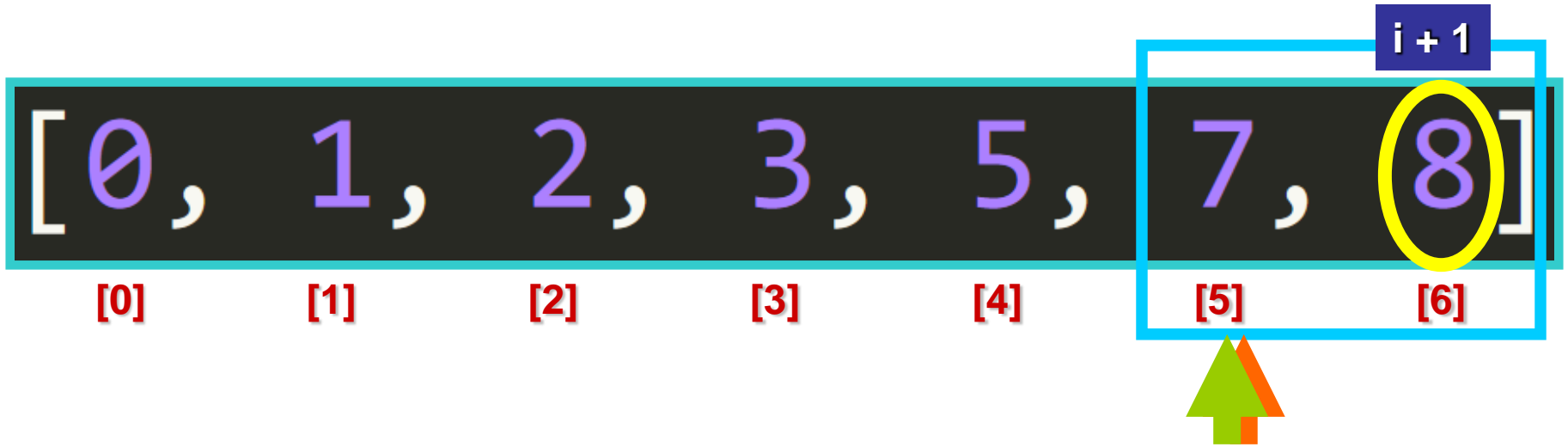


$i = 5$

$j = 5$



Quicksort

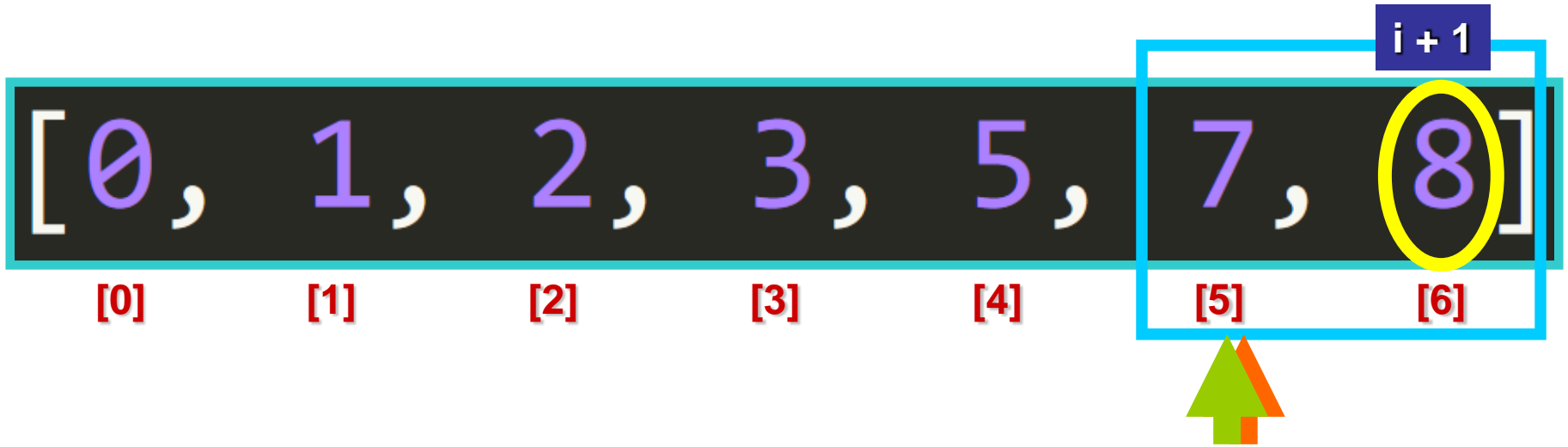


`i = 5`

`j = 5`



Quicksort



`i = 5`

`j = 5`



Quicksort

**Other recursive calls
do not make changes...**





Quicksort

[0, 1, 2, 3, 5, 7, 8]

Sorted



Time to Code!

