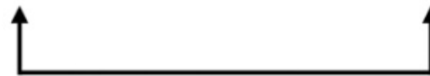


QUICKSORT

2	6	5	3	8	7	1	0
---	---	---	---	---	---	---	---

pivot

2	6	5	3	8	7	1	0
---	---	---	---	---	---	---	---



2	6	5	0	8	7	1	3
---	---	---	---	---	---	---	---



1. **itemFromLeft** that is larger than pivot

2. **itemFromRight** that is smaller than pivot

2	6	5	0	8	7	1	3
---	---	---	---	---	---	---	---

↑
itemFromLeft

↑
itemFromRight

↩ swap ↪

itemFromRight



2	1	5	0	8	7	6	3
---	---	---	---	---	---	---	---

↑ ↩ swap ↪
itemFromLeft

itemFromRight



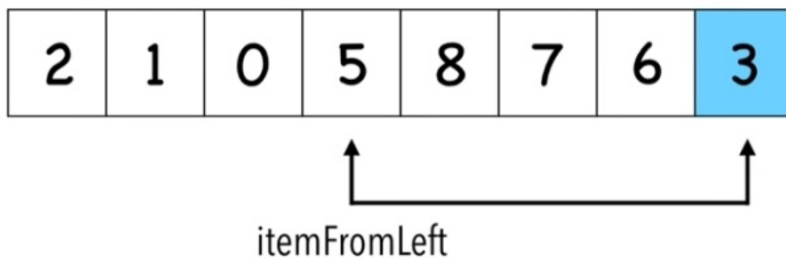
2	1	0	5	8	7	6	3
---	---	---	---	---	---	---	---

↑
itemFromLeft

Stop when index of **itemFromLeft** > index of **itemFromRight**

itemFromRight





Swap **itemFromLeft** and **pivot**

1. Correct position in final, sorted array
2. Items to the left are smaller
3. Items to the right are larger

2	1	0	3	8	7	6	5
---	---	---	---	---	---	---	---

and recursive

