

## 2.3 QUICK-SELECT

---



<http://algs4.cs.princeton.edu>

# Quick-select demo

---

Partition array so that:

- Entry  $a[j]$  is in place.
- No larger entry to the left of  $j$ .
- No smaller entry to the right of  $j$ .

Repeat in one subarray, depending on  $j$ ; finished when  $j$  equals  $k$ .

**select element of rank  $k = 5$**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
50	21	28	65	39	59	56	22	95	12	90	53	32	77	33

**$k = 5$**

# Quick-select demo

---

Partition array so that:

- Entry  $a[j]$  is in place.
- No larger entry to the left of  $j$ .
- No smaller entry to the right of  $j$ .

Repeat in one subarray, depending on  $j$ ; finished when  $j$  equals  $k$ .

partition on leftmost entry

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
50	21	28	65	39	59	56	22	95	12	90	53	32	77	33

$k = 5$

# Quick-select demo

---

Partition array so that:

- Entry  $a[j]$  is in place.
- No larger entry to the left of  $j$ .
- No smaller entry to the right of  $j$ .

Repeat in one subarray, depending on  $j$ ; finished when  $j$  equals  $k$ .

partitioned array

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
22	21	28	33	39	32	12	50	95	56	90	53	59	77	65

$k = 5$

# Quick-select demo

---

Partition array so that:

- Entry  $a[j]$  is in place.
- No larger entry to the left of  $j$ .
- No smaller entry to the right of  $j$ .

Repeat in one subarray, depending on  $j$ ; finished when  $j$  equals  $k$ .

can safely ignore right subarray

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
22	21	28	33	39	32	12	50	95	56	90	53	59	77	65

$k = 5$

# Quick-select demo

---

Partition array so that:

- Entry  $a[j]$  is in place.
- No larger entry to the left of  $j$ .
- No smaller entry to the right of  $j$ .

Repeat in one subarray, depending on  $j$ ; finished when  $j$  equals  $k$ .

partition on leftmost entry

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
22	21	28	33	39	32	12	50	95	56	90	53	59	77	65

$k = 5$

# Quick-select demo

---

Partition array so that:

- Entry  $a[j]$  is in place.
- No larger entry to the left of  $j$ .
- No smaller entry to the right of  $j$ .

Repeat in one subarray, depending on  $j$ ; finished when  $j$  equals  $k$ .

partitioned array

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
12	21	22	33	39	32	28	50	95	56	90	53	59	77	65

$k = 5$

# Quick-select demo

---

Partition array so that:

- Entry  $a[j]$  is in place.
- No larger entry to the left of  $j$ .
- No smaller entry to the right of  $j$ .

Repeat in one subarray, depending on  $j$ ; finished when  $j$  equals  $k$ .

can safely ignore left subarray

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
12	21	22	33	39	32	28	50	95	56	90	53	59	77	65

$k = 5$



# Quick-select demo

---

Partition array so that:

- Entry  $a[j]$  is in place.
- No larger entry to the left of  $j$ .
- No smaller entry to the right of  $j$ .

Repeat in one subarray, depending on  $j$ ; finished when  $j$  equals  $k$ .

partition on leftmost entry

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
12	21	22	33	39	32	28	50	95	56	90	53	59	77	65

$k = 5$

# Quick-select demo

---

Partition array so that:

- Entry  $a[j]$  is in place.
- No larger entry to the left of  $j$ .
- No smaller entry to the right of  $j$ .

Repeat in one subarray, depending on  $j$ ; finished when  $j$  equals  $k$ .

partitioned array

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
12	21	22	32	28	33	39	50	95	56	90	53	59	77	65

$k = 5$

# Quick-select demo

---

Partition array so that:

- Entry  $a[j]$  is in place.
- No larger entry to the left of  $j$ .
- No smaller entry to the right of  $j$ .

Repeat in one subarray, depending on  $j$ ; finished when  $j$  equals  $k$ .

**stop: partitioning item is at index  $k$**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
12	21	22	32	28	33	39	50	95	56	90	53	59	77	65

$k = 5$