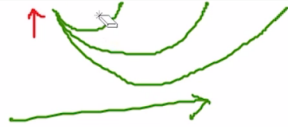


Selection Sort Explained

Length of array = 5

0	1	2	3	4
8	0	7	1	3



→ Not Sorted

Selection Sort Explained

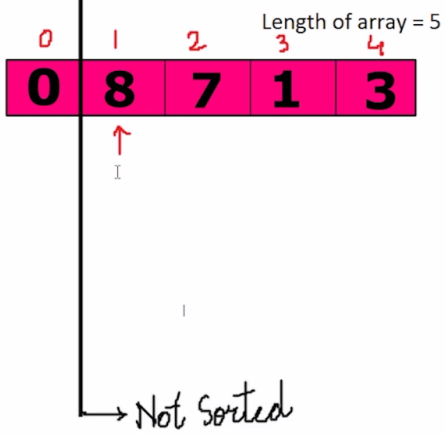
Length of array = 5

0	1	2	3	4
0	8	7	1	3

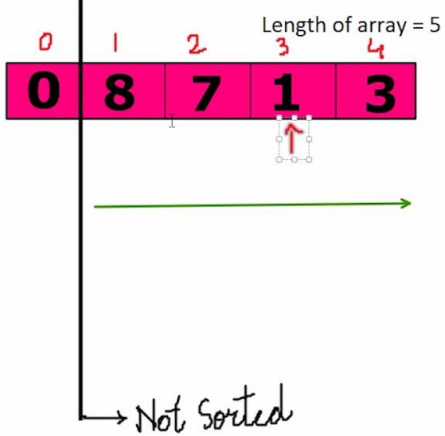


→ Not Sorted

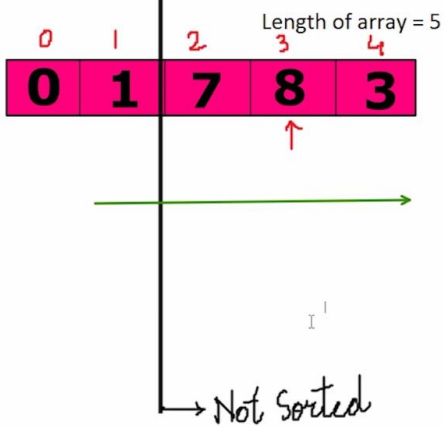
Selection Sort Explained



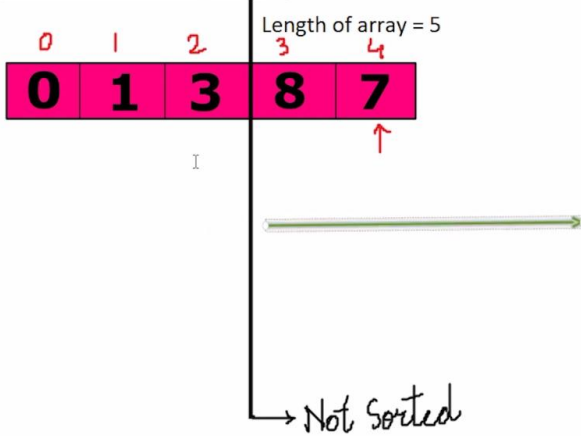
Selection Sort Explained



Selection Sort Explained



Selection Sort Explained



Selection Sort Explained

Length of array = 5

0	1	2	3	4
0	1	3	7	8



Not Sorted

Selection Sort Explained

0	1	2	3	4
8	0	7	1	3



Selection Sort Explained

0	1	2	3	4
8	0	7	1	3

1st pass

0	8	7	1	3
---	---	---	---	---

Selection Sort Explained

0	1	2	3	4
8	0	7	1	3

1st pass
0✓

0	8	7	1	3
---	---	---	---	---

2nd pass
0✓ 1✓

0	1	7	8	3
---	---	---	---	---

3rd pass
0✓ 1✓ 2✓

0	1	3	8	7
---	---	---	---	---

4th pass
0✓ 1✓ 2✓ 3✓

0	1	3	7	8
---	---	---	---	---

If length of the array = n

1. Number of passes required = $n - 1$
2. Total comparison = $\frac{n(n-1)}{2}$
3. Time complexity = $O(n^2)$
4. Total (max.) number of possible swaps = $n - 1$
5. It is **not adaptive**
6. It is **not stable** (as order not maintained)
7. Sorting in minimum number of swaps