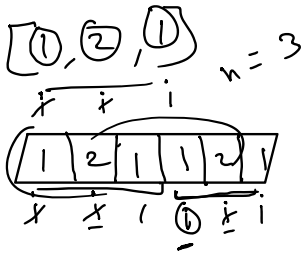
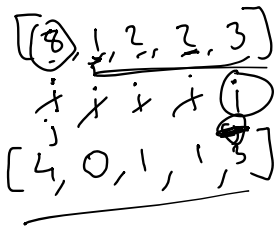
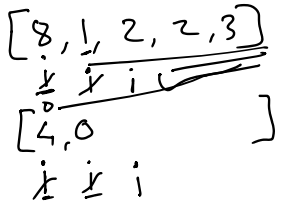
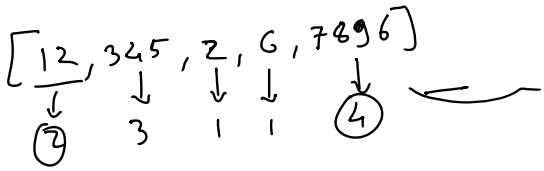




8





Sorting algorithm

↳ ascending order ⟶ small → large

↳ descending order ⟶ large → small

1. bubble sort
2. selection sort
3. insertion sort

[5, 1, 6, 2, 4, 3]

1st phase

1. [5, 1, 6, 2, 4, 3]
2. [1, 5, 6, 2, 4, 3]
3. [1, 5, 6, 2, 4, 3]
4. [1, 5, 2, 6, 4, 3]
5. [1, 5, 2, 4, 6, 3]
   [1, 5, 2, 4, 3, 6]

2nd phase

1. [1, 5, 2, 4, 3, 6]
2. [1, 5, 2, 4, 3, 6]
3. [1, 2, 5, 4, 3, 6]
4. [1, 2, 4, 5, 3, 6]
   [1, 2, 4, 3, 5, 6]

3rd phase

[1, 2, 4, 3, 5, 6]

4th phase

1. [1, 2, 3, 4, 5, 6]
2. [1, 2, 3, 4, 5, 6]
   [1, 2, 3, 4, 5, 6]

5th phase

1. [1, 2, 3, 4, 5, 6]
   [1, 2, 3, 4, 5, 6]

n − 1

n = 6

= n − i − 1

5 = 6 − 0 − 1

4 = 6 − 1 − 1

3 = 6 − 2 − 1

2 = 6 − 3 − 1

1 = 6 − 4 − 1

5 [1, 5, 2, 4, 3, 6]

[1, 5, 2, 4, 3, 6]

- 3rd phase

1 [1, 2, 4, 3, 5, 6]

2 [1, 2, 4, 3, 5, 6]

3 [1, 2, 4, 3, 5, 6]

[1, 2, 3, 4, 5, 6]

[1 2 3, 4, 5, 6]

3 - -

2 = 6 - 3 - 1

1 = 6 - 4 - 1

(arr[j] > arr[j+1]) }

}