# Bubble Sort

arr[] = {2, 6, 8, 3, 4}; n=5, j = n-1 = 4

① public void sort(int arr[], int n){

for(int i=0; i<n; i++){ (i=0) ✓ i=0 1

for( int j=0; j<n-1; j++){

if(arr[j] > arr[j+1]){

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

for(j=0; j<n; j++)

for(j=0; j<n-1; j++)

2 6 8 3 4 ×

j > j+1

2 6 3 8 4

2 6 3 4 8

2 3 6 4 8

2 3 4 6 8

② method-2

for(int i=0; i<n-1; i++){

for( int j=0; j<n-1-i; j++){ ✓

if(arr[j] > arr[j+1]){ ✓

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

Ascending order

Ascending order

if(arr[j] < arr[j+1]){

//some logic

}

Descending order

i=0 1 2 3 4

j=1 2

method-3

for(int i=0; i<n-1; i++){

for(int j=i+1; j<n; j++){

if(arr[i] > arr[j]){

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

6 3 5 4 2 → arr

3 6 5 4 2

2 6 5 4 3 → new Array

2 5 6 4 3

2 4 6 5 3

2 3 6 5 4

2 3 5 6 4

2 3 4 6 5

2 3 4 5 6

today covered topic

① BubbleSort

tomorrow topic

② Insertion Sort

$O(n^2)$ → worst case

$O(n)$ → best case

$O(n^2)$ → average case

$O(1)$ auxiliary → worst case space complexity.