Lec20

17 February 2026 10:39 AM

sorting algorithm
↳ ascending order → small → large
↳ descending order → large → large .

[5A, 3, 2, 5B]

[2, 3, 5A, 5B] → stable →

[2, 3, 5B, 5A] → unstable →

Bubble sort
selection sort
insertion sort

bubble sort

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

1st phase

5, 4, 2, 6, 1, 3 - 1
4, 5, 2, 6, 1, 3 - 2
4, 2, 5, 6, 1, 3 - 3
2, 5, 6, 1, 3 - 4
2, 5, 1, 6, 3 - 5
2, 5, 1, 3, 6

i = 0

2nd phase

4, 2, 5, 1, 3, 6 - 1
2, 4, 5, 1, 3, 6 - 2
2, 4, 5, 1, 3, 6 - 3
2, 4, 1, 5, 3, 6 - 4
2, 4, 1, 3, 5, 6

3rd phase

2, 4, 1, 3, 5, 6 - 1
2, 4, 1, 3, 5, 6 - 2
2, 1, 4, 3, 5, 6 - 3
2, 1, 3, 4, 5, 6

4th phase

2, 1, 3, 4, 5, 6 - 1
1, 2, 3, 4, 5, 6 - 2
1, 2, 3, 4, 5, 6

5th phase

1, 2, 3, 4, 5, 6 - 1
1, 2, 3, 4, 5, 6

n - 1

6 - 0 - 1

n - i - 1 = 5

j = 0 ; j < n - i - 1

```
for( i=0; i<n-1; i++){
    for(j=0; j<n-i-1; j++){
        if(arr[j] > arr[j+1]){
            int temp = arr[j]
            arr[j] = arr[j+1]
            arr[j+1] = temp;
```

}

{

}

selection sort

[7, 2, 4, 8, 10]

i j j j j j

[2, 7, 4, 8, 10]

i j j j j

[2, 4, 7, 8, 10]

i j j j

[2, 4, 7, 8, 10]

i j j

i j

[2, 4, 7, 8, 10] [0] = 7

current = ∅ 1 2 3

i = ∅ 1 2 3

j = i + 1

2 3 4

```
for(int i = 0; i < n-1; i++){
    int current = i
    for(int j = i+1; j < n; j++){
        if(arr[j] < arr[current]){
            current = j
        }
    }
    temp = arr[i]
    arr[i] = arr[current]
    arr[current] = temp
}
```

O(n^2)

O(n²)