

Selection Sort

List, Array

Algorithm

- ① find the minimum value in the list.
- ② swap it with the value in the current position.
- ③ Repeat this process for the all elements until the entire array is sorted.

array →

4	3	2	5	6
0	1	2	3	4

length = 5

for i = 0, 1, 2, 3, 4
for j → 2

Pass-1 → 4 3 2 5 6 → min = 2

Pass-2 → 2 3 4 5 6 → min = 3

Pass-3 → 2 3 4 5 6 → min = 4

Pass-4 → 2 3 4 5 6 → min = 5

Pass-5 → 2 3 4 5 6 → min = 6

asc/desc

① Ascending order

int arr[] = {4, 2, 3, 5, 6};

public void insertionSortAscendingOrder(int arr[]) {

int length = arr.length;

for (int i = 0; i < length; i++) {

int minIndex = i;

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

→ if (arr[j] < arr[minIndex]) {

minIndex = j;

} // end of if

} // end of inner for loop

int temp = arr[minIndex];

arr[minIndex] = arr[i];

arr[i] = temp;

} // end of the outer for loop.

minInd = 0

i = 0 ✓

j = 1

i

j

4

2

3

5

6

min = 1

i = 0, min = 1

minInd = 0

j = 1

j = 2

j = 3

j = 4

min = 3

i = 1

j = 2

j = 3

j = 4

min = 3

i = 2

j = 3

j = 4

min = 3

i = 3

j = 4

min = 3

i = 4

j = 4

min = 3

Selection Sort In Descending order

for (int i = 0; i < length; i++) {

int maxIndex = i;

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

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

maxIndex = j;

} // end of if condition.

} // end of inner for loop

int temp = arr[maxIndex];

arr[maxIndex] = arr[i];

arr[i] = temp;

} // end of outer for loop

Performance of Selection Sort

- ① Worst case time complexity = $O(n^2)$
- ② Best case time complexity = $O(n^2)$
- ③ Average case time complexity = $O(n^2)$
- ④ Worst case space complexity = $O(1)$ auxiliary

Covered topic (28-09-2023)

- ① Selection Sort Ascending order
- ② Selection Sort Descending order

Tomorrow topic (29-09-2023)

- ① Insertion Sort Ascending order
- ② Insertion Sort Descending order.