Assignment No: 03

Name of student:

Roll No:

Practical Batch:

Title of the Assignment: Implement Greedy search algorithm for application: Selection Sort.

Problem statement: Implement Selection Sort for Array.

Objective:

- To understand the concept of greedy search algorithm.
- To implement Selection Sort.

Theory

A greedy algorithm is an approach for solving a problem by selecting the best option available at the moment. It doesn't worry whether the current best result will bring the overall optimal result. The algorithm never reverses the earlier decision even if the choice is wrong. It works in a top-down approach.

Selection Sort

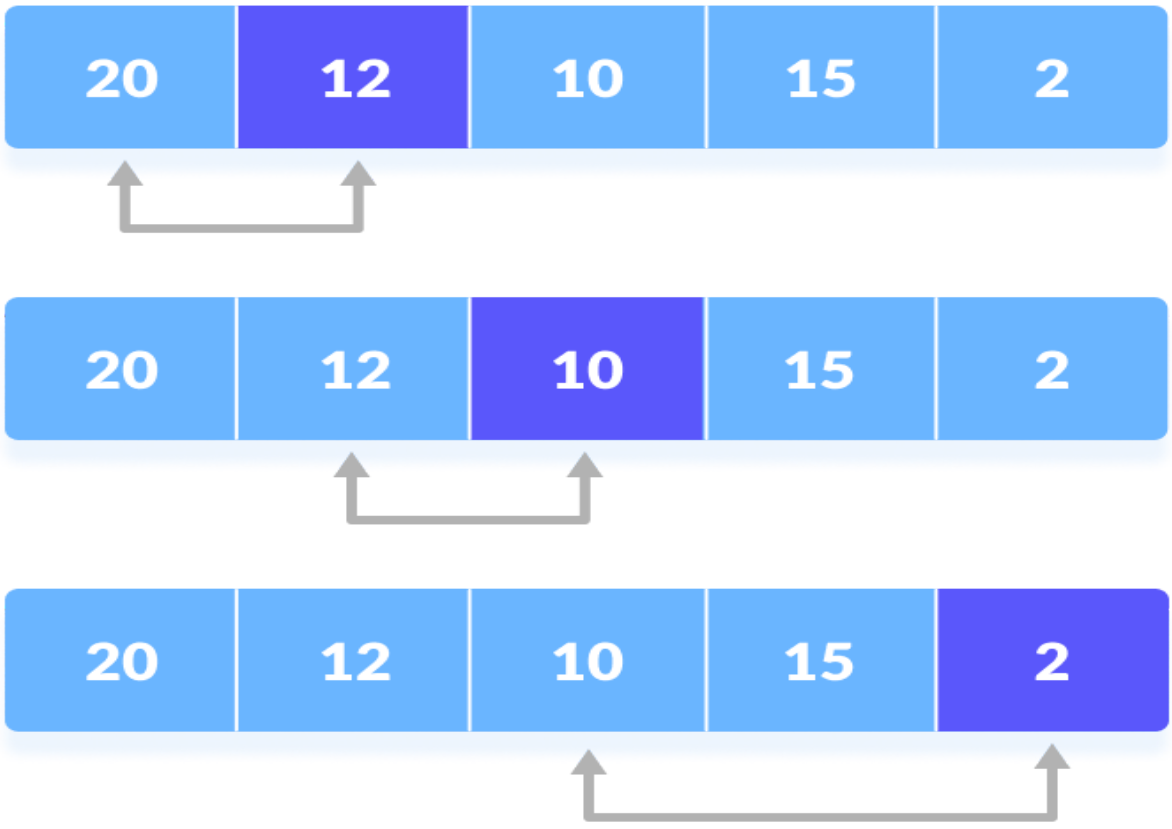
Selection sort is a sorting algorithm that selects the smallest element from an unsorted list in each iteration and places that element at the beginning of the unsorted list.

Working of Selection Sort

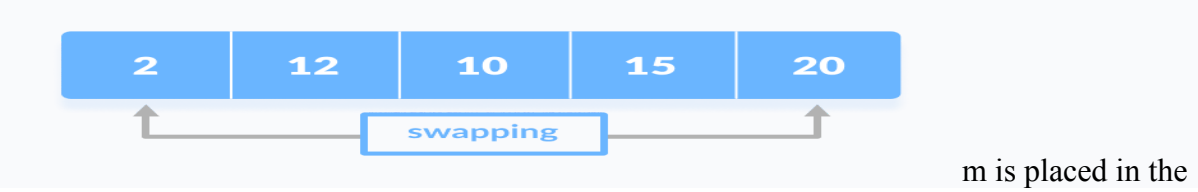
1. Set the first element as minimum.

2. Compare minimum with the second element. If the second element is smaller than minimum, assign the second element as minimum.

Compare minimum with the third element. Again, if the third element is smaller, then assign minimum to the third element otherwise do nothing. The process goes on until the last element.

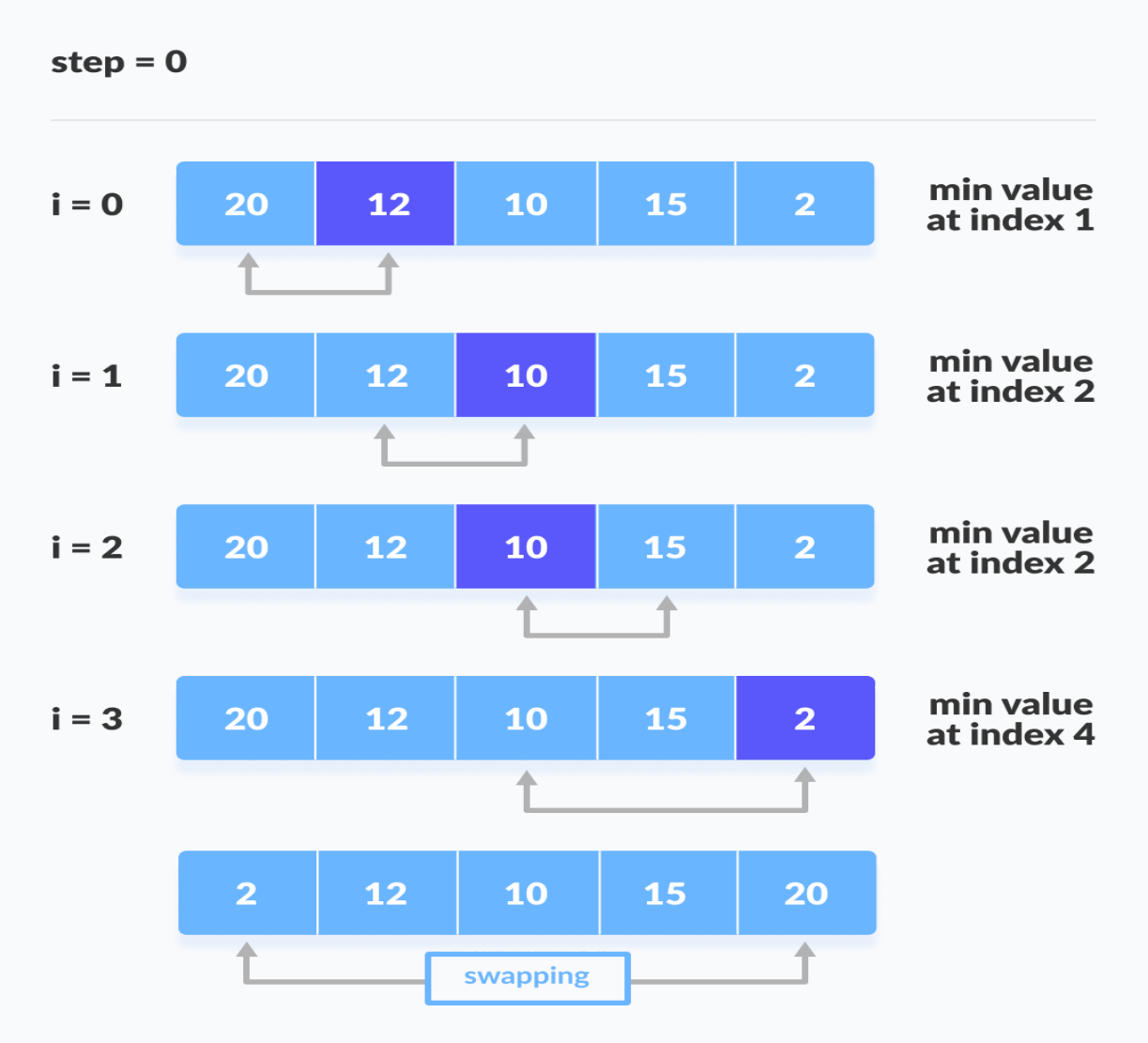


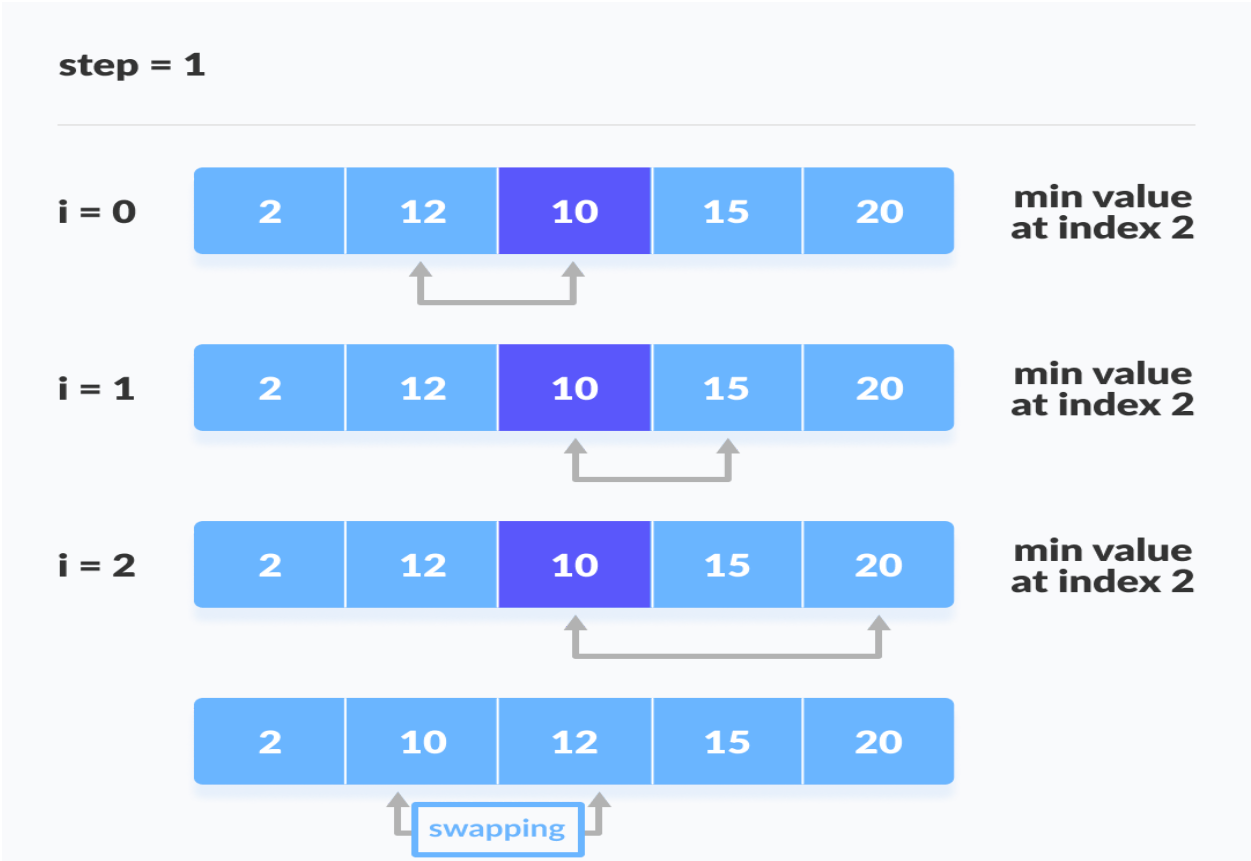
3. After each iteration, minimu

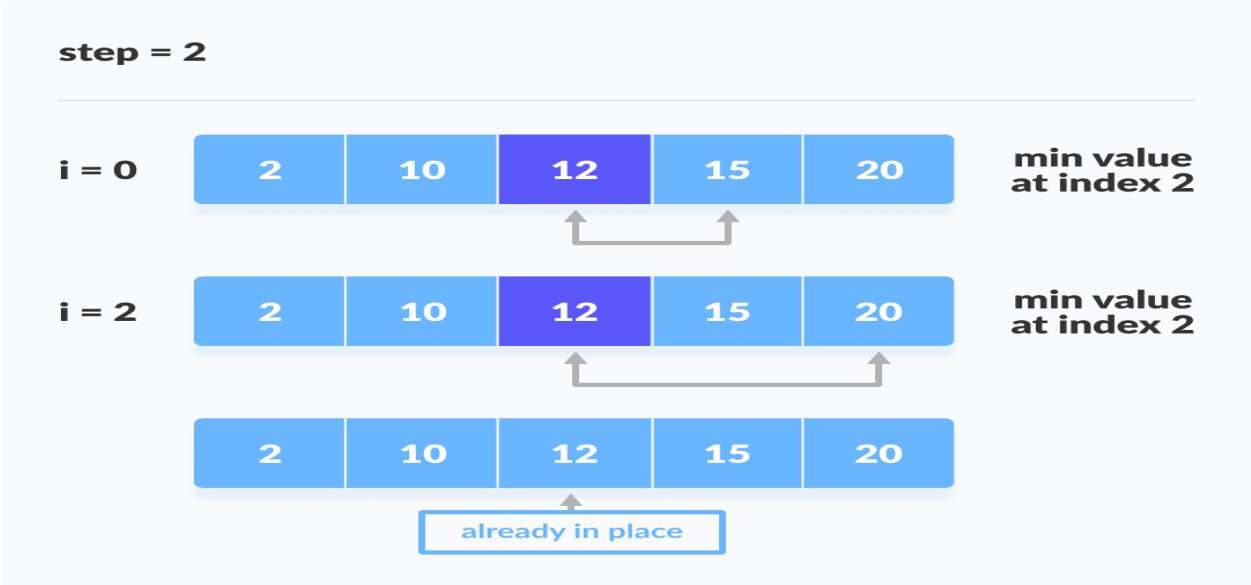


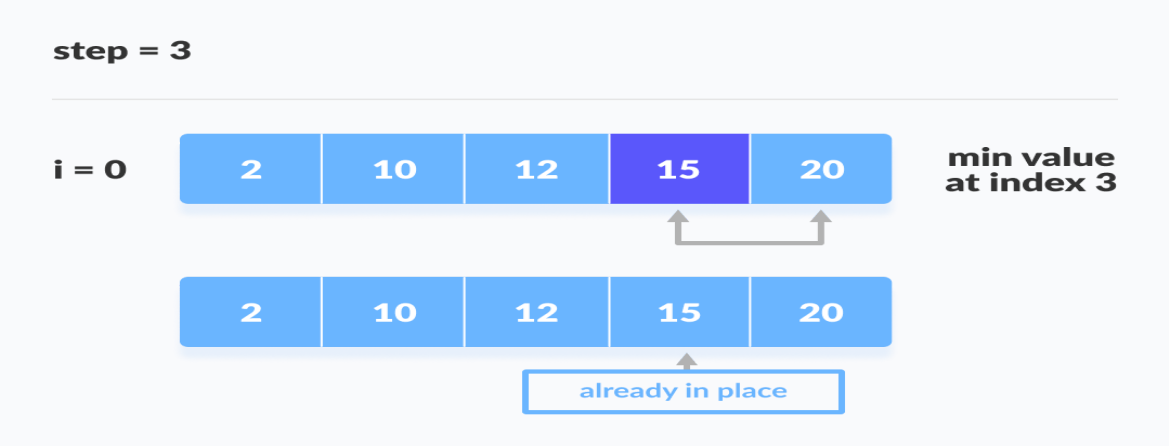
m is placed in the front of the unsorted list.

4. For each iteration, indexing starts from the first unsorted element. Step 1 to 3 are repeated until all the elements are placed at their correct positions.




step = 1
i = 0
2
12
10
15
20
min value at index 2
i = 1
2
12
10
15
20
min value at index 2
i = 2
2
12
10
15
20
min value at index 2
2
10
12
15
20
swapping


step = 2
i = 0
2
10
12
15
20
min value at index 2
i = 2
2
10
12
15
20
min value at index 2
2
10
12
15
20
already in place



Selection Sort Algorithm

selectionSort(array, size)

repeat (size - 1) times

set the first unsorted element as the minimum

for each of the unsorted elements

if element < currentMinimum

set element as new minimum

swap minimum with first unsorted position

end selectionSort

Time Complexities:

Worst Case Complexity: O(n2)

If we want to sort in ascending order and the array is in descending order then, the worst case occurs.

Best Case Complexity: O(n2)

It occurs when the array is already sorted

Average Case Complexity: O(n2)

It occurs when the elements of the array are in jumbled order (neither ascending nor descending).

Conclusion

We have implemented Selection sort