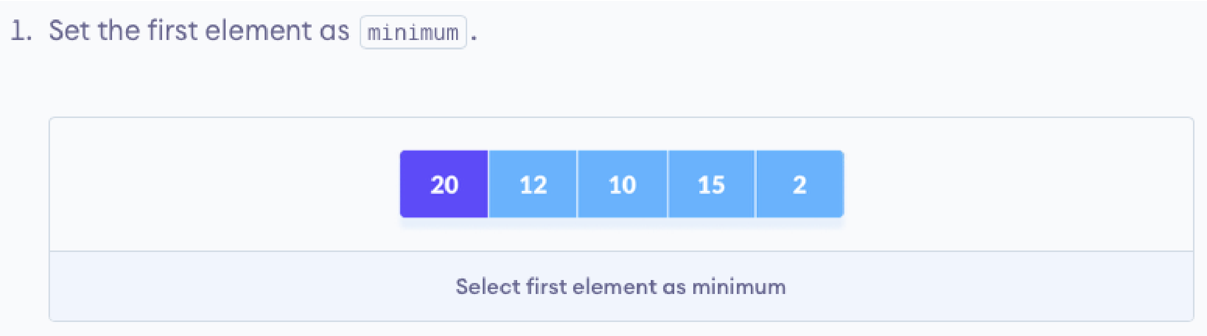
Selection Sort

> 💡 **Selection Sort** is a comparison-based sorting algorithm. It sorts an array by repeatedly selecting the **smallest (or largest)**element from the unsorted portion and swapping it with the first unsorted element. This process continues until the entire array is sorted.

How it works:

1. Go through the array to find the lowest value.
2. Move the lowest value to the front of the unsorted part of the array.
3. Go through the array again as many times as there are values in the array.

Algorithm steps

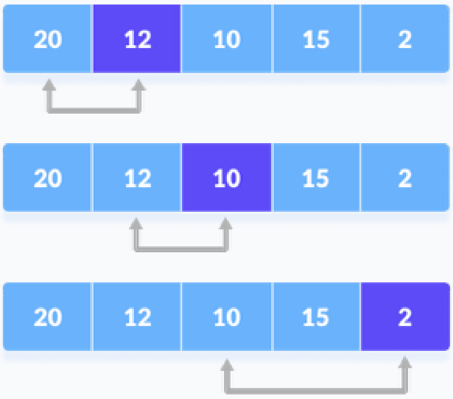


1. Set the first element as `minimum`.

Select first element as minimum

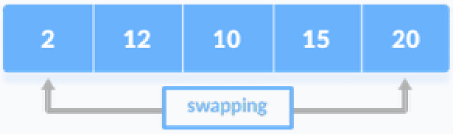
2. Compare `minimum` with the second element. If the second element is smaller than `minimum`, assign the second element as `minimum`.

   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.

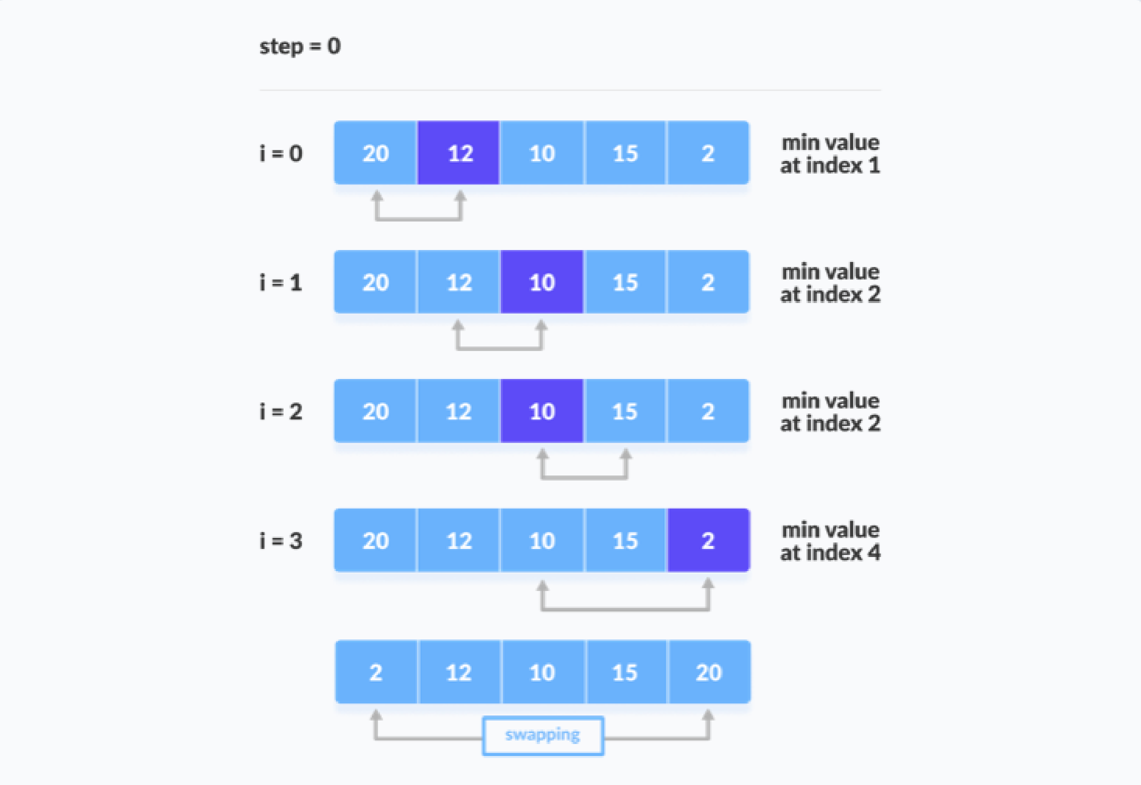
   Compare minimum with the remaining elements

3. After each iteration, `minimum` is placed in the front of the unsorted list.

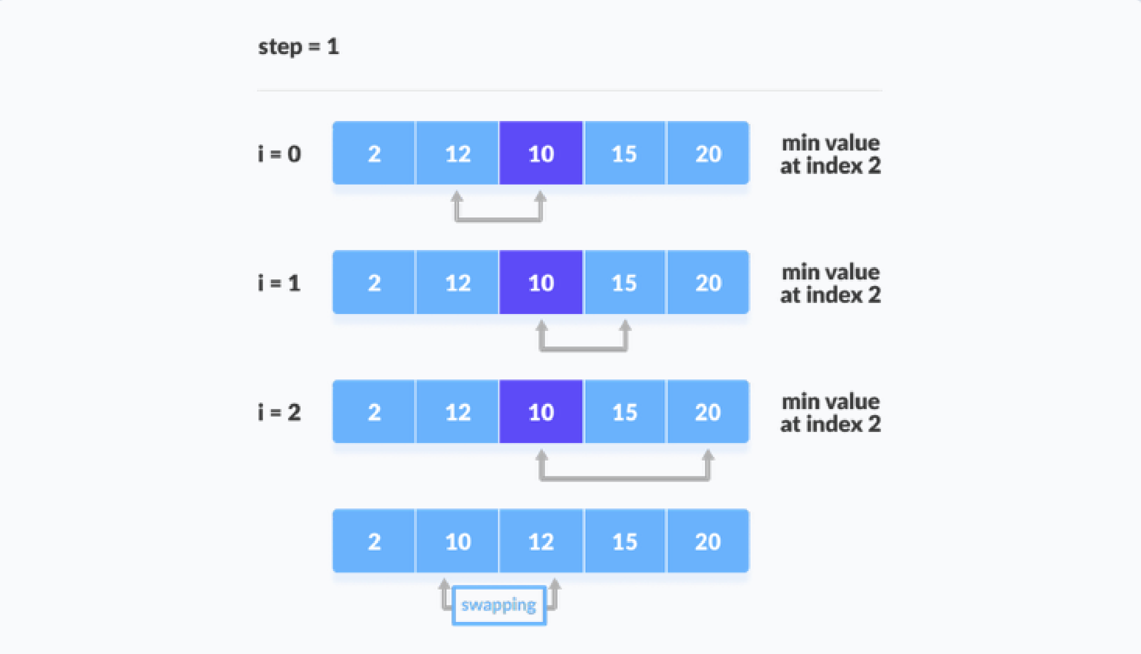
   

   Swap the first with minimum

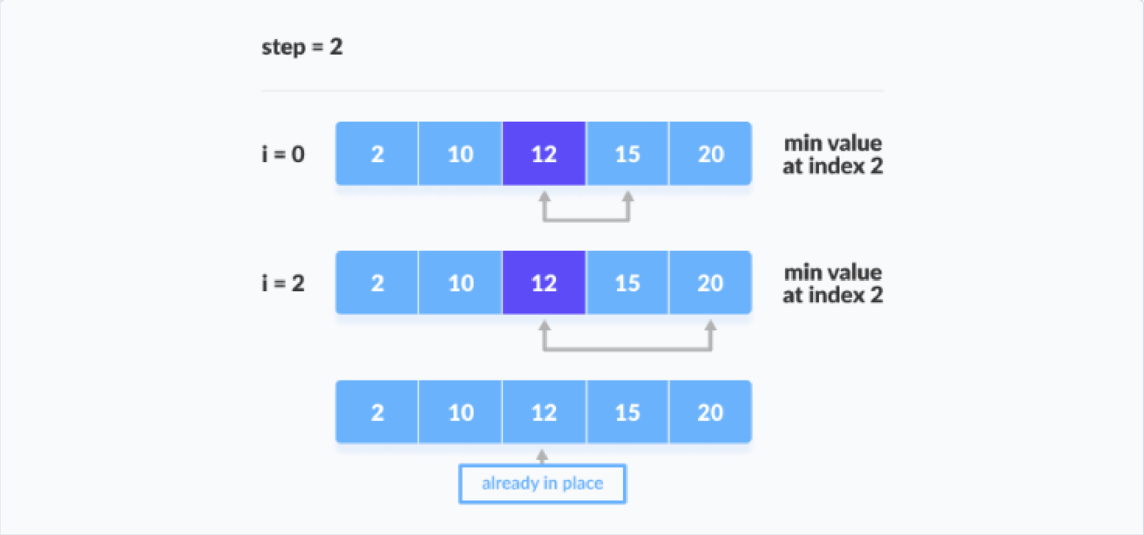
4. For each iteration, indexing starts from the first unsorted element. Step 1 to 3 are repeated until all the elements are placed at their correct positions.



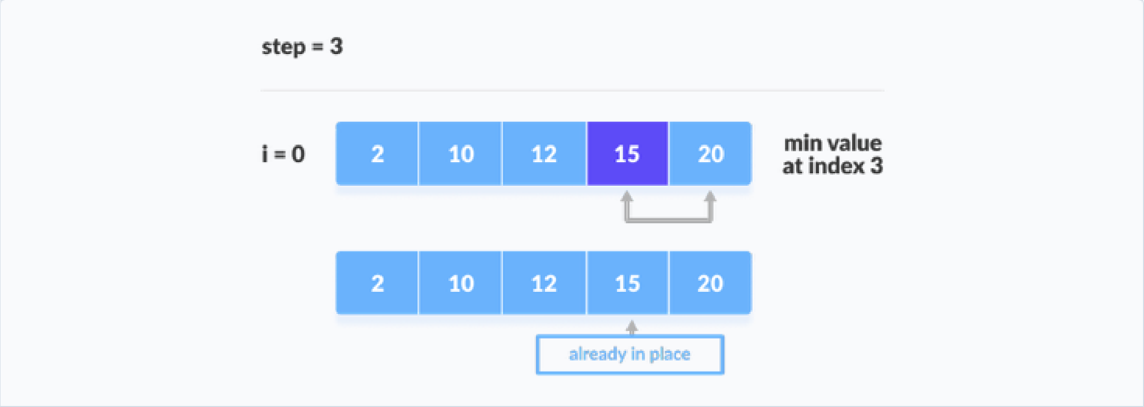
The first iteration



The second iteration



The third iteration



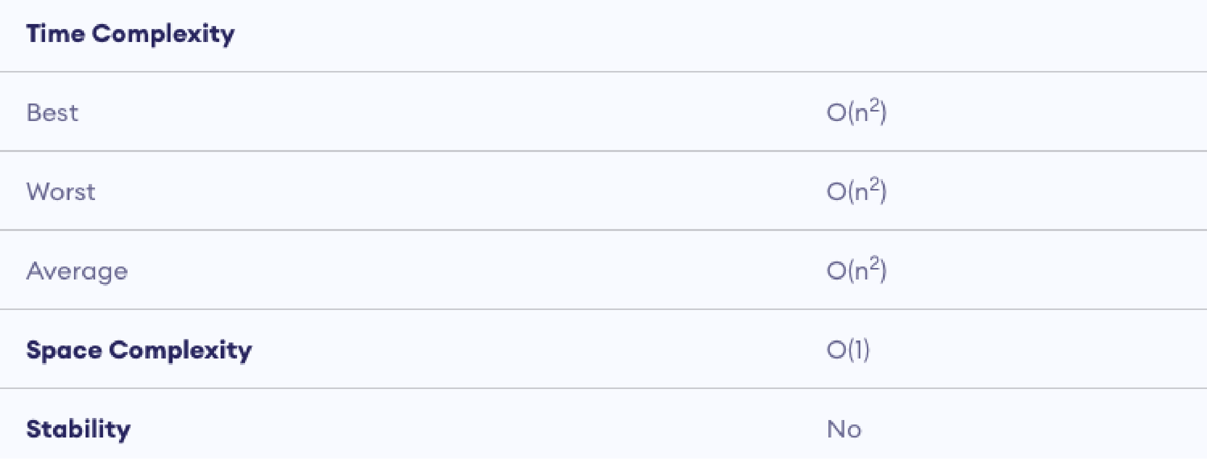
The fourth iteration

Pseudocode

```
selectionSort(array, size)
  for i from 0 to size - 1 do
    set i as the index of the current minimum
    for j from i + 1 to size - 1 do
      if array[j] < array[current minimum]
        set j as the new current minimum index
    if current minimum is not i
```

```
      swap array[i] with array[current minimum]
end selectionSort
```

Analysis



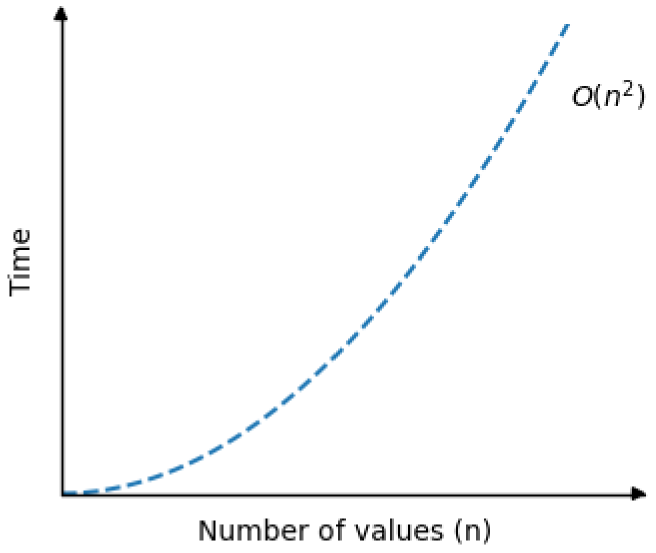
Time Complexity	
Best	$O(n^2)$
Worst	$O(n^2)$
Average	$O(n^2)$
Space Complexity	$O(1)$
Stability	No

Complexity = `O(n2)`

Also, we can analyze the complexity by simply observing the number of loops. There are 2 loops so the complexity is `n*n = n2`.

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).



Advantages of Selection Sort

- Easy to understand and implement, making it ideal for teaching basic sorting concepts.
- Requires only a constant O(1) extra memory space.
- It requires less number of swaps (or memory writes) compared to many other standard algorithms. Only cycle sort beats it in terms of memory writes. Therefore it can be simple algorithm choice when memory writes are costly.

Disadvantages of the Selection Sort

- Selection sort has a time complexity of O(n^2) makes it slower compared to algorithms like Quick Sort and Merge Sort.
- Does not maintain the relative order of equal elements which means it is not stable.

Selection Sort Applications

The selection sort is used when

- a small list is to be sorted

- cost of swapping does not matter
- checking of all the elements is compulsory
- cost of writing to a memory matters like in flash memory (number of writes/swaps is `O(n)` as compared to `O(n2)` of bubble sort)

📌 Java implementation can be found under Implementation_Java folder

📚 References

Selection Sort (With Code in Python/C++/Java/C)

Selection Sort is an algorithm that works by selecting the smallest element from the array and putting it at its correct position and then selecting the second smallest element and putting it at its correct position and so on (for ascending order). In this tutorial, you will understand the working of selection

https://www.programiz.com/dsa/selection-sort

W3Schools.com

W3Schools offers free online tutorials, references and exercises in all the major languages of the web. Covering popular subjects like HTML, CSS, JavaScript, Python, SQL,

https://www.w3schools.com/dsa/dsa_algo_selectionsort.php

Selection Sort - GeeksforGeeks

Selection Sort is a comparison-based sorting algorithm that repeatedly selects the smallest element from the unsorted portion of an array and swaps it with the first unsorted

https://www.geeksforgeeks.org/selection-sort-algorithm-2/

✍🏻 Author → Serhat Kumas

https://www.linkedin.com/in/serhatkumas/

SerhatKumas - Overview

Computer engineering student who loves coding in different fields instead of focusing on a one spesific area. - SerhatKumas

https://github.com/SerhatKumas