

Theory: Selection sort

🕒 11 minutes 11 / 11 problems solved

Start practicing

884 users solved this topic. Latest completion was about 2 hours ago.

Selection sort is a simple sorting algorithm that performs an in-place sorting.

Let's see how the algorithm works for sorting an array in the ascending order. First, it finds the smallest element in the whole array and swaps it with the element in the first position; then it finds the second smallest element and swaps it with the element in the second position. It continues doing that until the whole array is sorted.

If we want to sort the array in the descending order, we should find the largest element instead of the smallest one.

If n is the length of an input array, the algorithm has asymptotic time complexity $O(n^2)$ in the worst and average cases in terms of the number of comparisons. This makes the algorithm inefficient for sorting large arrays. The algorithm finds the minimum/maximum element $n - 1$ times.

The basic implementation of the algorithm is **unstable**, but it can be modified to be **stable**.

Current topic:

✓ [Selection sort](#) ...

Table of contents:

- [1 Selection sort](#)
- [§1. Example](#)
- [§2. Double selection sort](#)
- [Feedback & Comments](#)

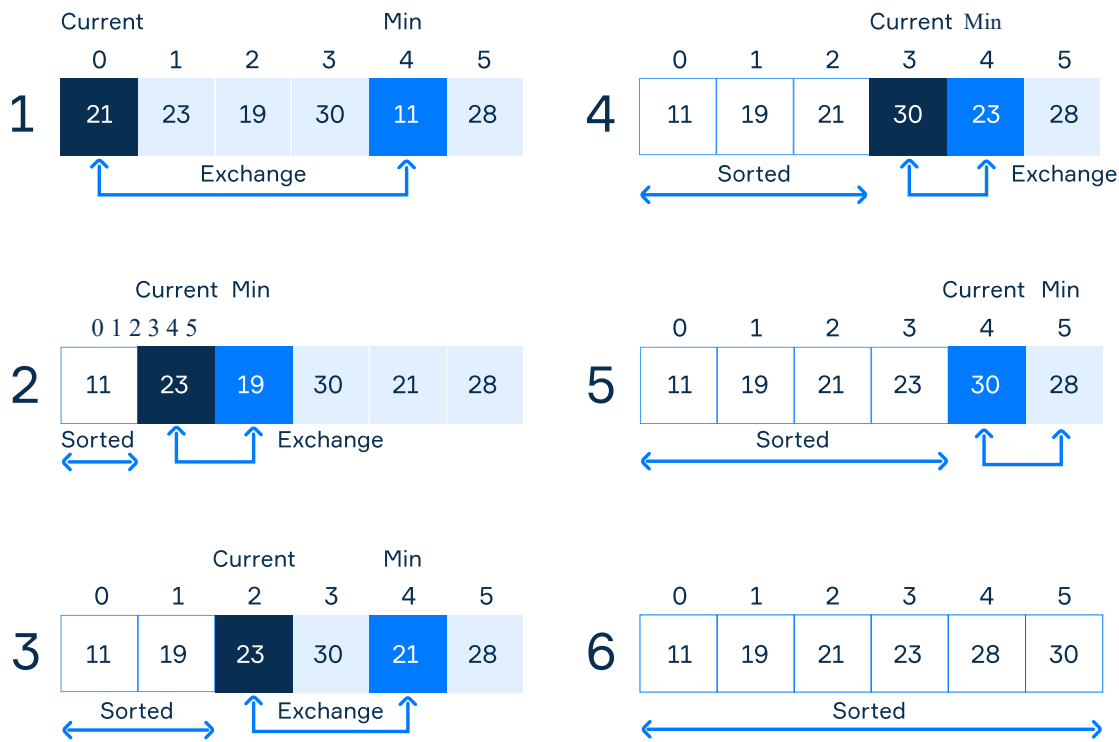
§1. Example

Suppose we have an unsorted array of integers and we should sort it in the ascending order.

0	1	2	3	4	5
21	23	19	30	11	28

This array has six elements, the first element has the index 0, the last one has the index 5.

The following image illustrates how the sorting algorithm works:



Here are some explanations:

- 1) We find the min number in the whole array (11) and swap it with the first element (21). Now, the first element belongs to the sorted subarray.
- 2) We find the min number in the unsorted subarray (19) and swap the number with the second element. Now, the first and second elements belong to the sorted subarray.

3) We find the min number in the unsorted subarray (21) and swap it with the third element (23). Now, the first three elements belong to the sorted subarray.

4-6) We repeat the same process until the whole array is sorted.


As you can see, the algorithm is quite simple. It never changes the already sorted subarray.

You may also check out a [visualization](#) of Selection Sort to better understand it.

§2. Double selection sort

The bidirectional variant of selection sort finds both the minimum and maximum values in the array at every pass. It reduces the number of scans of the array. The algorithm divides the array into three subarrays: 1) sorted minimums; 2) unsorted; 3) sorted maximums. The algorithm has the same time complexity as the basic algorithm: $O(n^2)$.

 Report a typo

 Thanks for your feedback!

Start practicing

[Comments \(1\)](#)

[Hints \(2\)](#)

[Useful links \(0\)](#)

[Show discussion](#)