Theory: Selection sort in Java

24 minutes 0 / 4 problems solved

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

Suppose we sort an array in ascending order. The algorithm works as follows. First, it finds the smallest element in the whole array and exchanges it with the element at the first position, then finds the second smallest element and exchanges it with the element at the second position, and continues in this way until the whole array is sorted.

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

If n is the length of an input array, the algorithm has asymptotic time complexity $\mathrm{O}(n^2)$ in the worst and average cases in terms of the number of comparisons. It 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**.

§1. Implementation in Java

Selection sort algorithm can be easily implemented in Java. The following implementation has two loops. In the outer loop, we exchange array elements. In the nested loop, we find the index of the next smallest element to be exchanged.

```
public static int[] selectionSort(int[] array) {
    for (int i = 0; i < array.length - 1; i++) {
        int index = i; // the index of the found min

        /* Iterating over the unsorted subarray to find the min */
        for (int j = i + 1; j < array.length; j++) {
            if (array[j] < array[index]) {
                index = j;
            }

        }



        /* Exchanging the found min and the current element */

        int min = array[index];

        array[index] = array[i];

        array[i] = min;

    }


    return array;

}
```

Let's test the method passing different arrays:

```
selectionSort(new int[] { 21, 23, 19, 30, 11, 28 }); // { 11, 19, 21, 23, 28, 30 }
selectionSort(new int[] { 30, 28, 23, 21, 19, 11 }); // { 11, 19, 21, 23, 28, 30 }
```

You can start the implemented algorithm in the debug mode to understand it better.

Current topic:

Selection sort in Java

Topic depends on:

- ✓ Selection sort
- ✗ Algorithms in Java

Table of contents:

37 users liked this theory. **1** didn't like it. **What about you?**

Start practicing

Comments (5) Hints (0) Useful links (0) Show discussion