6 LIVE EVENTS

All Tracks > Algorithms > Sorting > Selection Sort

# Algorithms

Solve any problem to achieve a rank

View Leaderboard

Topics: Selection Sort

## Selection Sort

TUTORIAL PROBLEMS VISUALIZER BETA

The Selection sort algorithm is based on the idea of finding the minimum or maximum element in an unsorted array and then putting it in its correct position in a sorted array.

Assume that the array $A = [7, 5, 4, 2]$ needs to be sorted in ascending order.

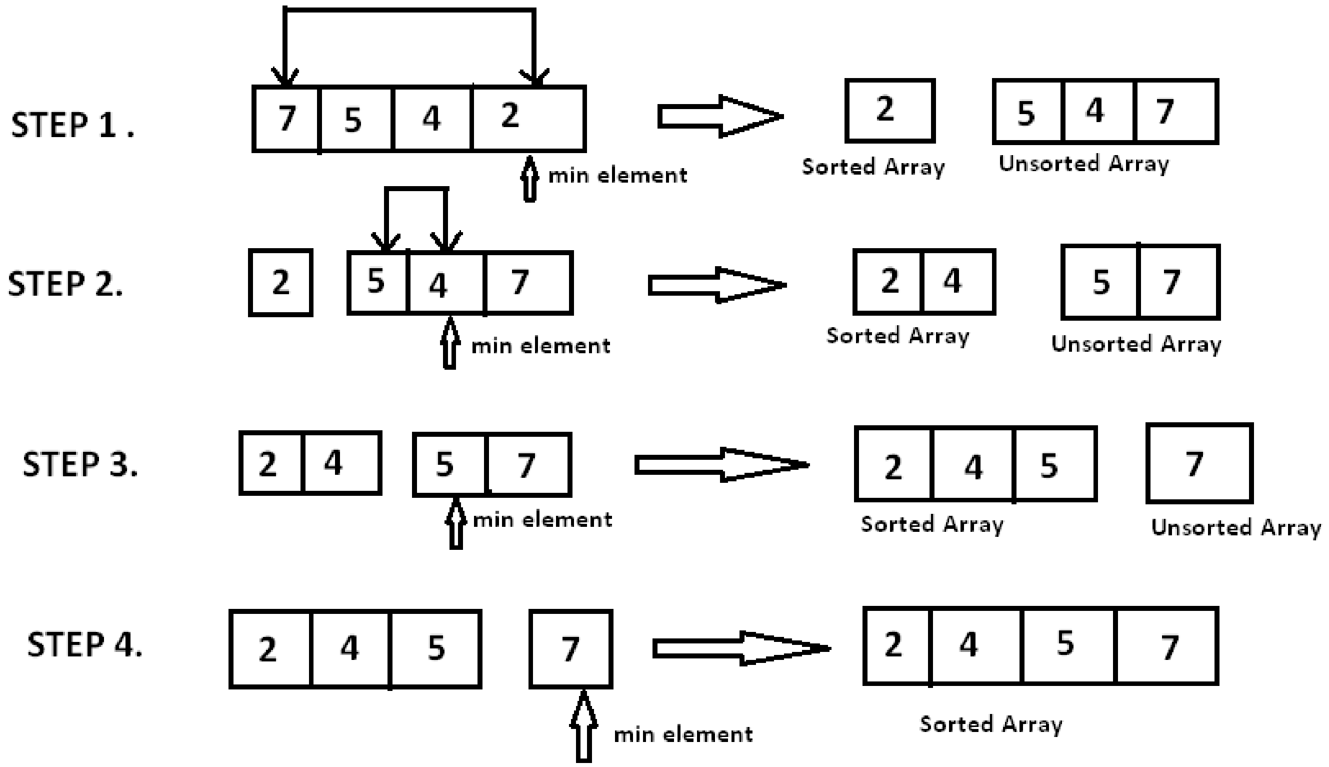
The minimum element in the array i.e. $2$ is searched for and then swapped with the element that is currently located at the first position, i.e. $7$. Now the minimum element in the remaining unsorted array is searched for and put in the second position, and so on.

Let’s take a look at the implementation.

```
void selection_sort (int A[ ], int n) {
        // temporary variable to store the position of minimum element

        int minimum;
        // reduces the effective size of the array by one in  each iteration.

        for(int i = 0; i < n-1 ; i++)  {

           // assuming the first element to be the minimum of the unsorted array .
             minimum = i ;

          // gives the effective size of the unsorted  array .

            for(int j = i+1; j < n ; j++ ) {
                if(A[ j ] < A[ minimum ])  {                //finds the minimum element
                minimum = j ;
```

```
              }
           }
       // putting minimum element on its proper position.
       swap ( A[ minimum ], A[ i ]) ;
     }
}
```

At $i^{th}$ iteration, elements from position $0$ to $i-1$ will be sorted.



Time Complexity:

To find the minimum element from the array of $N$ elements, $N-1$ comparisons are required. After putting the minimum element in its proper position, the size of an unsorted array reduces to $N-1$ and then $N-2$ comparisons are required to find the minimum in the unsorted array.

Therefore $(N-1)$ + $(N-2)$ + $.......$ + $1$ = $(N \cdot (N-1))/2$ comparisons and $N$ swaps result in the overall complexity of $O(N^2)$.

*Contributed by: Anand Jaisingh*

Did you find this tutorial helpful? YES NO

## TEST YOUR UNDERSTANDING

?

## Selection Sort

- Consider an Array *a* of size *N*
- Iterate from 1 to *N*
- In $i^{th}$ iteration select the $i^{th}$ minimum and swap it with $a[i]$

6 LIVE EVENTS

You are given an array *a*, size of the array *N* and an integer *x*. Follow the above algorithm and print the state of the array after *x* iterations have been performed.

**Input Format**

The first line contains two integer *N* and *x* denoting the size of the array and the steps of the above algorithm to be performed respectively. The next line contains *N* space separated integers denoting the elements of the array.

**Output Format**

Print *N* space separated integers denoting the state of the array after *x* steps

**Constraints**

$$1 \leq N \leq 100$$

$$1 \leq a[i] \leq 100$$

$$1 \leq x \leq N$$

SAMPLE INPUT

```
5 2
1 2 3 4 5
```

SAMPLE OUTPUT

```
1 2 3 4 5
```

Enter your code or Upload your code as file. Save C (gcc 5.4.0) ▾

```
/*
// Sample code to perform I/O:
#include <stdio.h>

int main(){
    int num;
    scanf("%d", &num);                          // Reading input from STDIN
    printf("Input number is %d.\n", num);       // Writing output to STDOUT
}

// Warning: Printing unwanted or ill-formatted data to output will cause the test cases to fail
*/

// Write your code here
```

?