

Selection Sort Algorithm

Consider sorting n numbers stored in array A by first finding the smallest element of A and exchanging it with element in $A[1]$. Then find the second smallest element of A , and exchange it with $A[2]$. Continue this manner for the first $n - 1$ elements of A .

You can find the implementation [here](#) or go to the next url: https://github.com/DiegoMendezMedina/C_Algorithms/tree/master/Sort/Selection_Sort/implementations.

Pseudocode

SELECTION-SORT(A)

1. **for** $i = 1$ **to** $n-1$
2. $\text{smallest} = A[i]$
3. $k = i$
4. **for** $j = i$ **to** n
5. **if** $A[j] < \text{smallest}$
6. $\text{smallest} = A[j]$
7. $k = j$
8. **if** $i \neq k$
9. $\text{change}(a, i, k)$

CHANGE(A, b, c)

1. $\text{aux} = A[b]$
2. $A[b] = A[c]$
3. $A[C] = \text{aux}$

Proof

Loop invariant:

At the start of each iteration of the **for** loop (lines 1-9), $A[1..i]$ is sorted. i.e the i -th element of the array is greater for every previous numbers on the array.

On the second **for** loop (lines 4-7), the smallest number in $A[i..n]$ is found **if** it's different from the one in the i -th position there's a *change*. So now $A[1..i+1]$ is also sorted.

Initialization:

When $i = 1$ there are no previous **i** values.

The smallest number in $A[1..n]$ is found and is changed with the one in $A[1]$. Now $A[1,2]$ is sorted.

Maintenance:

There's another iteration which means that $A[1..i]$ is sorted.

The smallest number in $A[i..n]$ is found and is changed with the one in $A[i]$. Now $A[1..i+1]$ is sorted.

Termination:

$A[i...n-1]$ is sorted and $A[n]$ is greater than $A[n-1]$, hence $A[1...n]$ is sorted.