geeksforgeeks.org

Selection Sort - GeeksforGeeks

4 minutes

The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning. The algorithm maintains two subarrays in a given array.

1) The subarray which is already sorted.
2) Remaining subarray which is unsorted.

In every iteration of selection sort, the minimum element (considering ascending order) from the unsorted subarray is picked and moved to the sorted subarray.

Following example explains the above steps:

```
arr[] = 64 25 12 22 11

// Find the minimum element in arr[0...4]
// and place it at beginning
11 25 12 22 64

// Find the minimum element in arr[1...4]
// and place it at beginning of arr[1...4]
11 12 25 22 64

// Find the minimum element in arr[2...4]
// and place it at beginning of arr[2...4]
11 12 22 25 64
```

```
// Find the minimum element in arr[3...4]
// and place it at beginning of arr[3...4]
11 12 22 25 64
```

- C/C++
- Python
- Java

C/C++

```
#include <stdio.h>

void swap(int *xp, int *yp)

{

    int temp = *xp;

    *xp = *yp;

    *yp = temp;

}

void selectionSort(int arr[], int n)

{

    int i, j, min_idx;

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

    {

        min_idx = i;

        for (j = i+1; j < n; j++)

          if (arr[j] < arr[min_idx])

            min_idx = j;

        swap(&arr[min_idx], &arr[i]);
```

```
    }
}
void printArray(int arr[], int size)
{
    int i;
    for (i=0; i < size; i++)
        printf("%d ", arr[i]);
    printf("\n");
}
int main()
{
    int arr[] = {64, 25, 12, 22, 11};
    int n = sizeof(arr)/sizeof(arr[0]);
    selectionSort(arr, n);
    printf("Sorted array: \n");
    printArray(arr, n);
    return 0;
}
```

Python

```
import sys
A = [64, 25, 12, 22, 11]
for i in range(len(A)):
    min_idx = i
```

```
    for j in range(i+1, len(A)):
        if A[min_idx] > A[j]:
            min_idx = j
    A[i], A[min_idx] = A[min_idx], A[i]
print ("Sorted array")
for i in range(len(A)):
    print("%d" %A[i]),
```

Java

```
class SelectionSort
{
    void sort(int arr[])
    {
        int n = arr.length;
        for (int i = 0; i < n-1; i++)
        {
            int min_idx = i;
            for (int j = i+1; j < n; j++)
                if (arr[j] < arr[min_idx])
                    min_idx = j;
            int temp = arr[min_idx];
            arr[min_idx] = arr[i];
            arr[i] = temp;
        }
```

```
    }
    void printArray(int arr[])
    {
        int n = arr.length;
        for (int i=0; i<n; ++i)
            System.out.print(arr[i]+" ");
        System.out.println();
    }
    public static void main(String args[])
    {
        SelectionSort ob = new SelectionSort();
        int arr[] = {64,25,12,22,11};
        ob.sort(arr);
        System.out.println("Sorted array");
        ob.printArray(arr);
    }
}
```

Output:

```
Sorted array:
11 12 22 25 64
```

Time Complexity: $O(n^2)$ as there are two nested loops.

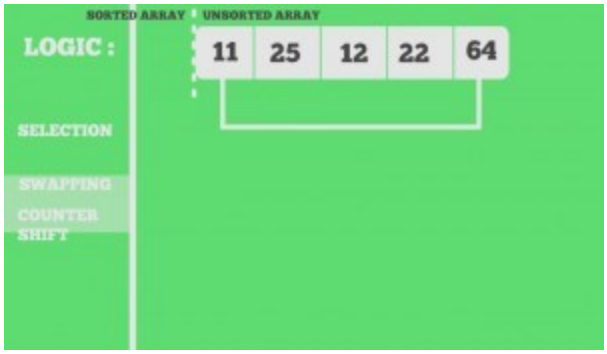
Auxiliary Space: O(1)

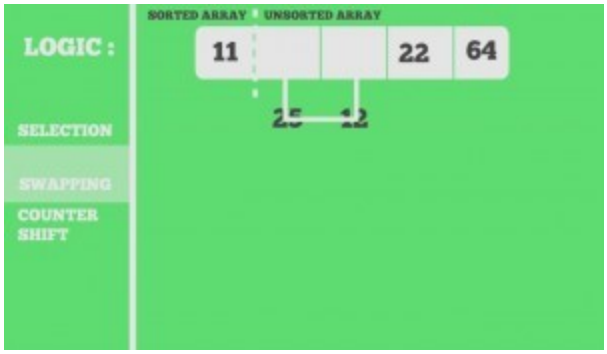
The good thing about selection sort is it never makes more than O(n) swaps and can be useful when memory write is a costly operation.

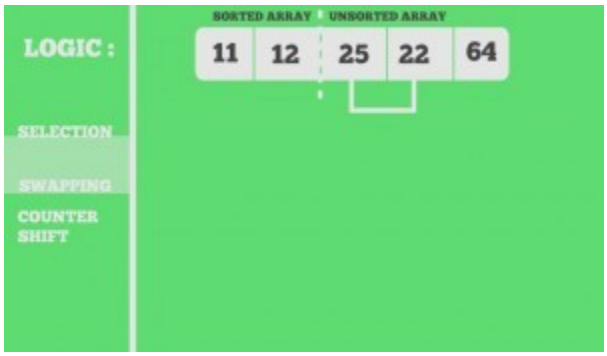
Exercise :

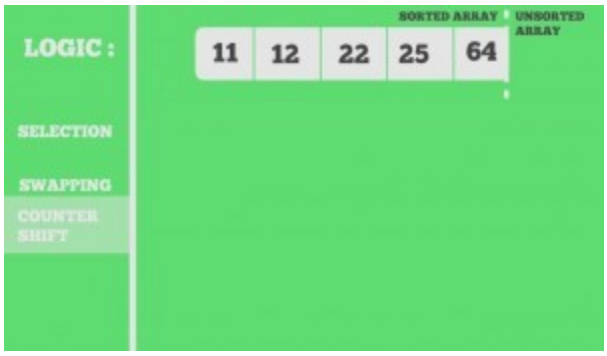
Sort an array of strings using Selection Sort

Snapshots:









Quiz on Selection Sort

Other Sorting Algorithms on GeeksforGeeks/GeeksQuiz:

- Bubble Sort
- Insertion Sort
- Merge Sort
- Heap Sort
- QuickSort
- Radix Sort
- Counting Sort
- Bucket Sort
- ShellSort

Coding practice for sorting.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.