**Selection Sort**

Selection sort is a sorting algorithm that selects the smallest element from an unsorted list in each iteration and places that element at the beginning of the unsorted list.

**Selection Sort Pseudocode:**

procedure selection sort

list : array of items

n : size of list

for i = 1 to n - 1

min = i

for j = i+1 to n

if list[j] < list[min] then

min = j;

end if

end for

if indexMin != i then

swap list[min] and list[i]

end if

end for

end procedure

**Complexities:** Time Complexity: Best – O(n2), Average – O(n2), Worst – O(n2)

Space Complexity: O(1)

Stability: Yes

**Applications:** The selection sort is used when

* a small list is to be sorted
* the cost of swapping does not matter
* checking all the elements is compulsory
* cost of writing to a memory matter like in flash memory (number of writes/swaps is O(n) as compared to O(n2) of bubble sort)

**Source Code:**

using System;

namespace SelectionSort

{

class Program

{

static void Main(String[] args)

{

Input();

}

static void Input()

{

Console.Write("Enter Number of Items: ");

int noOfItems = Convert.ToInt32(Console.ReadLine());

int[] itemsList = new int[noOfItems];

Console.Write("Enter Items: ");

for (int i = 0; i < noOfItems; i++)

{

itemsList[i] = Convert.ToInt32(Console.ReadLine());

}

Console.Write("For ascending write 'a' or descending wirte 'd': ");

char order = Convert.ToChar(Console.ReadLine());

SelectionSort(itemsList, order);

}

static void SelectionSort(int[] itemsList, char order)

{

for (int i = 0; i < itemsList.Length-1; i++)

{

int minOrMaxIndex = i;

for(int j = i+1; j < itemsList.Length; j++)

{

if (order == 'a')

{

if (itemsList[minOrMaxIndex] > itemsList[j])

{

minOrMaxIndex = j;

}

}

else if(order == 'd')

{

if (itemsList[minOrMaxIndex] < itemsList[j])

{

minOrMaxIndex = j;

}

}

}

if(minOrMaxIndex != i)

{

int temp = itemsList[i];

itemsList[i] = itemsList[minOrMaxIndex];

itemsList[minOrMaxIndex] = temp;

}

}

Output(itemsList);

}

static void Output(int[] itemList)

{

Console.Write("After sorting: ");

for (int i = 0; i < itemList.Length; i++)

{

Console.Write($"{itemList[i]}\t");

}

}

}}