**Bubble Sort**

Bubble sort is a sorting algorithm that compares two adjacent elements and swaps them until they are in the intended order.

**Bubble Pseudocode:**

procedure bubbleSort(list : array of items)

loop = list.count;

for i = 0 to loop-1 do:

swapped = false

for j = 0 to loop-1 do:

if list[j] > list[j+1] then

swap(list[j], list[j+1])

swapped = true

end if

end for

if(not swapped) then

break

end if

end for

end procedure return list

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

Space Complexity: O(1)

Stability: Yes

**Applications:** Bubble sort is used if complexity does not matter and short or simple code is preferred.

**Source Code:**

using System;

namespace BubbleSort

{

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 write 'd': ");

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

BubbleSort(itemsList, order);

}

static void BubbleSort(int[] itemList, char order)

{

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

{

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

{

if(order == 'a')

{

if (itemList[j] > itemList[j + 1])

{

int temp = itemList[j];

itemList[j] = itemList[j + 1];

itemList[j + 1] = temp;

}

}

else if (order == 'd')

{

if (itemList[j] < itemList[j + 1])

{

int temp = itemList[j];

itemList[j] = itemList[j + 1];

itemList[j + 1] = temp;

}

}

}

}

Output(itemList);

}

static void Output(int[] itemList)

{

Console.Write("After sorting: ");

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

{

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

}

}

}

}