Danish Khalid

Project: Sorting Algorithms

01 134141 029

CS 5(A)

Introduction

This project contains commonly used sorting algorithms. These algorithms when executed show their inner workings by displaying each iteration which is used to sort an array.

Following sorting algorithms are used:

* Insertion Sort
* Selection Sort
* Bubble Sort
* Quick Sort
* Merge Sort

These algorithms have their own advantages and disadvantages considering various scenarios. The iterations in a sorting algorithm help us understand their processing and memory demand and give us a generalized insight into their time complexity in practical use.

How to Use

Usage is simple: you can input any numbers with limit up to 10. They can be negative or zeroes. Sorting is limited to integers only so no double or floating point numbers. If you enter a number with decimal points; the digits beyond decimal points will not be considered in the software’s calculations as they will be discarded.

Code

# Main Menu

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Project\_X\_\_\_Sorting\_Algorithms

{

public partial class Menu : Form

{

public Menu()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

this.Hide();

InsertionSort im = new InsertionSort();

im.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Hide();

SelectionSort ss = new SelectionSort();

ss.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

this.Hide();

QuickSort qs = new QuickSort();

qs.Show();

}

private void button4\_Click(object sender, EventArgs e)

{

this.Hide();

BubbleSort bs = new BubbleSort();

bs.Show();

}

private void button5\_Click(object sender, EventArgs e)

{

this.Hide();

MergeSort ms = new MergeSort();

ms.Show();

}

private void Menu\_FormClosing(object sender, FormClosingEventArgs e)

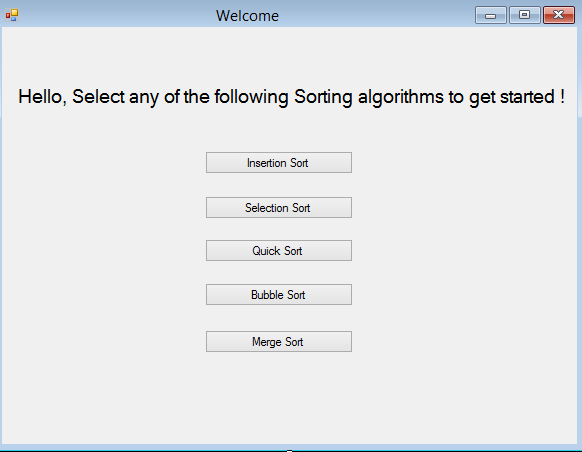
{

Application.Exit();

}

}

}



# Insertion Sort

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

namespace Project\_X\_\_\_Sorting\_Algorithms

{

public partial class InsertionSort : Form

{

private int[] arr = new int[10];

List<List<TextBox>> lists = new List<List<TextBox>>();

List<TextBox> textBoxes = new List<TextBox>();

List<TextBox> one = new List<TextBox>();

List<TextBox> two = new List<TextBox>();

List<TextBox> three = new List<TextBox>();

List<TextBox> four = new List<TextBox>();

List<TextBox> five = new List<TextBox>();

List<TextBox> six = new List<TextBox>();

List<TextBox> seven = new List<TextBox>();

List<TextBox> eight = new List<TextBox>();

List<TextBox> nine = new List<TextBox>();

List<TextBox> ten = new List<TextBox>();

public InsertionSort()

{

InitializeComponent();

textBoxes = new List<TextBox>

{

textBox1,

textBox2,

textBox3,

textBox4,

textBox5,

textBox6,

textBox7,

textBox8,

textBox9,

textBox10

};

lists = new List<List<TextBox>>

{

one,

two,

three,

four,

five,

six,

seven,

eight,

nine,

ten

};

}

private void button1\_Click(object sender, EventArgs e)

{

for (int i = 0; i < 10; i++)

{

try

{

arr[i] = int.Parse(textBoxes[i].Text);

}

catch (Exception)

{

arr[i] = 0;

}

}

insertion(arr);

for (int i = 0; i < 10; i++)

textBoxes[i].Text = "";

}

private void insertion(int[] arr)

{

int i, j, value;

for (i = 0; i < arr.Length; i++)

{

value = arr[i];

for (j = i - 1; j >= 0 && arr[j] > value; j--)

{

arr[j + 1] = arr[j];

}

arr[j + 1] = value;

List<TextBox> temp = lists[i];

for (int x = 0; x < 10; x++)

temp[x].Text = arr[x].ToString();

}

}

private void InsertionMenu\_Load(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

this.Hide();

Menu M = new Menu();

M.Show();

}

private void InsertionSort\_FormClosing(object sender, FormClosingEventArgs e)

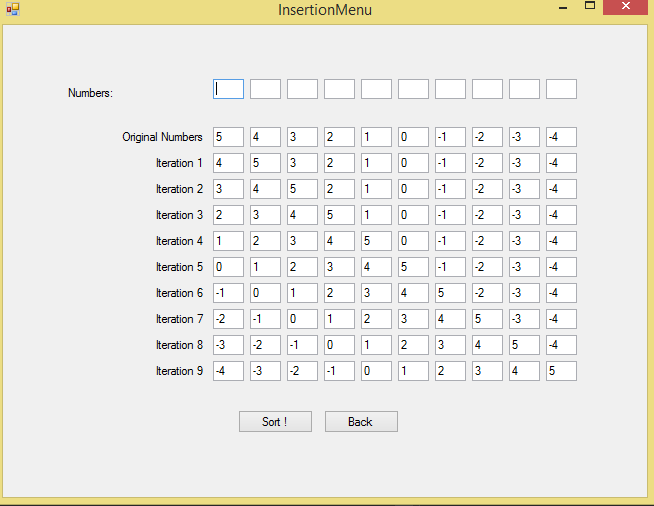
{

Application.Exit();

}

}

}



# Selection Sort

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Project\_X\_\_\_Sorting\_Algorithms

{

public partial class SelectionSort : Form

{

private int[] arr = new int[10];

List<TextBox> textBoxes = new List<TextBox>();

List<List<TextBox>> lists = new List<List<TextBox>>();

List<TextBox> one = new List<TextBox>();

List<TextBox> two = new List<TextBox>();

List<TextBox> three = new List<TextBox>();

List<TextBox> four = new List<TextBox>();

List<TextBox> five = new List<TextBox>();

List<TextBox> six = new List<TextBox>();

List<TextBox> seven = new List<TextBox>();

List<TextBox> eight = new List<TextBox>();

List<TextBox> nine = new List<TextBox>();

List<TextBox> ten = new List<TextBox>();

public SelectionSort()

{

InitializeComponent();

textBoxes = new List<TextBox>

{

textBox1,

textBox2,

textBox3,

textBox4,

textBox5,

textBox6,

textBox7,

textBox8,

textBox9,

textBox10

};

lists = new List<List<TextBox>>

{

one,

two,

three,

four,

five,

six,

seven,

eight,

nine,

ten

};

}

private void button1\_Click(object sender, EventArgs e)

{

for (int i = 0; i < 10; i++)

{

try

{

arr[i] = int.Parse(textBoxes[i].Text);

}

catch (Exception)

{

arr[i] = 0;

}

}

selection(arr);

for (int i = 0; i < 10; i++)

textBoxes[i].Text = arr[i].ToString();

}

private void selection(int[] arr)

{

int i, j, T, min;

for (i = 0; i < arr.Length; i++)

{

min = i;

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

{

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

{

min = j;

}

}

T = arr[min];

arr[min] = arr[i];

arr[i] = T;

List<TextBox> temp = lists[i];

for (int x = 0; x < 10; x++)

temp[x].Text = arr[x].ToString();

}

}

private void button2\_Click(object sender, EventArgs e)

{

this.Hide();

Menu M = new Menu();

M.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

for (int i = 0; i < 10; i++)

textBoxes[i].Text = "";

}

private void SelectionSort\_FormClosing(object sender, FormClosingEventArgs e)

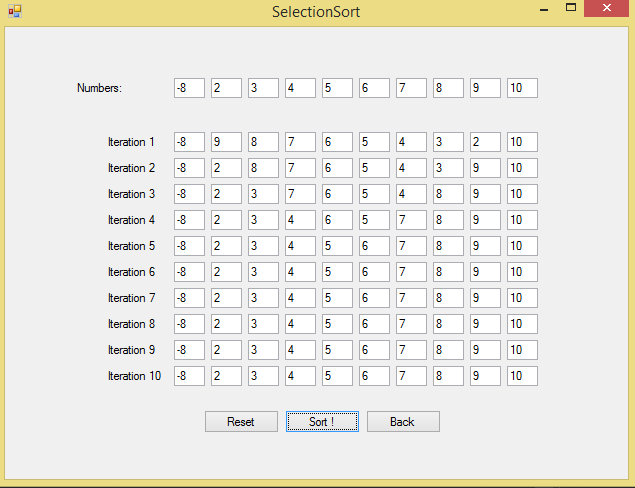
{

Application.Exit();

}

}

}



# Quicksort

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Project\_X\_\_\_Sorting\_Algorithms

{

public partial class QuickSort : Form

{

private int[] arr = new int[10];

List<TextBox> textBoxes = new List<TextBox>();

List<List<TextBox>> lists = new List<List<TextBox>>();

List<TextBox> one = new List<TextBox>();

List<TextBox> two = new List<TextBox>();

List<TextBox> three = new List<TextBox>();

List<TextBox> four = new List<TextBox>();

List<TextBox> five = new List<TextBox>();

List<TextBox> six = new List<TextBox>();

List<TextBox> seven = new List<TextBox>();

List<TextBox> eight = new List<TextBox>();

List<TextBox> nine = new List<TextBox>();

List<TextBox> ten = new List<TextBox>();

public QuickSort()

{

InitializeComponent();

textBoxes = new List<TextBox>

{

textBox1,

textBox2,

textBox3,

textBox4,

textBox5,

textBox6,

textBox7,

textBox8,

textBox9,

textBox10

};

lists = new List<List<TextBox>>

{

one,

two,

three,

four,

five,

six,

seven,

eight,

nine,

ten

};

}

private void button1\_Click(object sender, EventArgs e)

{

for (int i = 0; i < 10; i++)

{

try

{

arr[i] = int.Parse(textBoxes[i].Text);

}

catch (Exception)

{

arr[i] = 0;

}

}

for(int i = 0; i < 10; i++)

for(int j = 0; j < 10; j++)

{

if(arr[i] == arr[j])

{

MessageBox.Show("Duplicates Not Allowed !");

for (int k = 0; k < 10; k++)

textBoxes[k].Text = "";

return;

}

}

quick(arr, 0, arr.Length - 1);

}

static public int Partition(int[] numbers, int left, int right)

{

int pivot = numbers[left];

while (true)

{

while (numbers[left] < pivot)

left++;

while (numbers[right] > pivot)

right--;

if (left < right)

{

int temp = numbers[right];

numbers[right] = numbers[left];

numbers[left] = temp;

}

else

{

return right;

}

}

}

struct QuickPosInfo

{

public int left;

public int right;

};

public void quick(int[] numbers, int left, int right)

{

int y = 0;

if (left >= right)

return; // Invalid index range

List<QuickPosInfo> list = new List<QuickPosInfo>();

QuickPosInfo info;

info.left = left;

info.right = right;

list.Insert(list.Count, info);

while (true)

{

if (list.Count == 0)

break;

left = list[0].left;

right = list[0].right;

list.RemoveAt(0);

int pivot = Partition(numbers, left, right);

if (pivot > 1)

{

info.left = left;

info.right = pivot - 1;

list.Insert(list.Count, info);

}

if (pivot + 1 < right)

{

info.left = pivot + 1;

info.right = right;

list.Insert(list.Count, info);

}

if (y < 10)

{

List<TextBox> temp = lists[y];

for (int x = 0; x < 10; x++)

temp[x].Text = arr[x].ToString();

}

y++;

}

}

private void button2\_Click(object sender, EventArgs e)

{

this.Hide();

Menu M = new Menu();

M.Show();

}

private void QuickSort\_FormClosing(object sender, FormClosingEventArgs e)

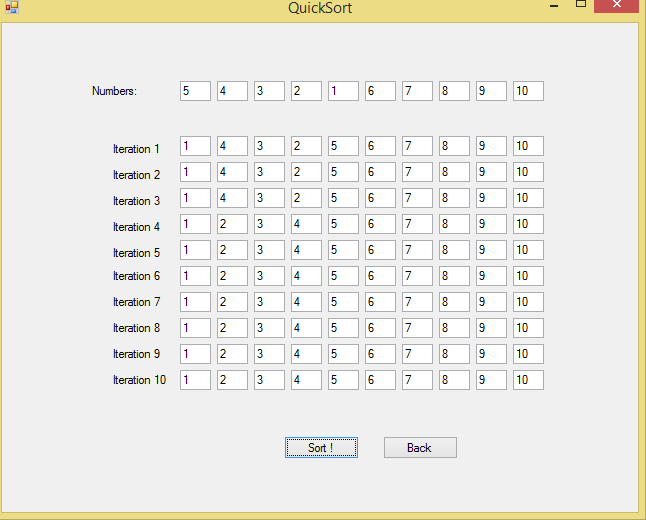
{

Application.Exit();

}

}

}



# BubbleSort

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Project\_X\_\_\_Sorting\_Algorithms

{

public partial class BubbleSort : Form

{

private int[] arr = new int[10];

List<TextBox> textBoxes = new List<TextBox>();

List<List<TextBox>> lists = new List<List<TextBox>>();

List<TextBox> one = new List<TextBox>();

List<TextBox> two = new List<TextBox>();

List<TextBox> three = new List<TextBox>();

List<TextBox> four = new List<TextBox>();

List<TextBox> five = new List<TextBox>();

List<TextBox> six = new List<TextBox>();

List<TextBox> seven = new List<TextBox>();

List<TextBox> eight = new List<TextBox>();

List<TextBox> nine = new List<TextBox>();

List<TextBox> ten = new List<TextBox>();

public BubbleSort()

{

InitializeComponent();

textBoxes = new List<TextBox>

{

textBox1,

textBox2,

textBox3,

textBox4,

textBox5,

textBox6,

textBox7,

textBox8,

textBox9,

textBox10

};

lists = new List<List<TextBox>>

{

one,

two,

three,

four,

five,

six,

seven,

eight,

nine,

ten

};

}

private void button1\_Click(object sender, EventArgs e)

{

for (int k = 0; k < 10; k++)

{

List<TextBox> gg = lists[k];

for (int x = 0; x < 10; x++)

gg[x].Text = "";

}

for (int i = 0; i < 10; i++)

{

try

{

arr[i] = int.Parse(textBoxes[i].Text);

}

catch(Exception)

{

arr[i] = 0;

}

}

bubble(arr);

}

private void bubble(int[] arr)

{

int length = arr.Length;

bool isSorted = false;

int y = 0;

while(!isSorted)

{

isSorted = true;

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

{

if(arr[i] > arr[i + 1])

{

int temp = arr[i];

arr[i] = arr[i + 1];

arr[i + 1] = temp;

isSorted = false;

}

}

length--;

List<TextBox> gg = lists[y];

for (int x = 0; x < 10; x++)

gg[x].Text = arr[x].ToString();

y++;

}

}

private void button2\_Click(object sender, EventArgs e)

{

this.Hide();

Menu M = new Menu();

M.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

for (int k = 0; k < 10; k++)

{

textBoxes[k].Text = "";

List<TextBox> gg = lists[k];

for (int x = 0; x < 10; x++)

gg[x].Text = "";

}

}

private void BubbleSort\_FormClosing(object sender, FormClosingEventArgs e)

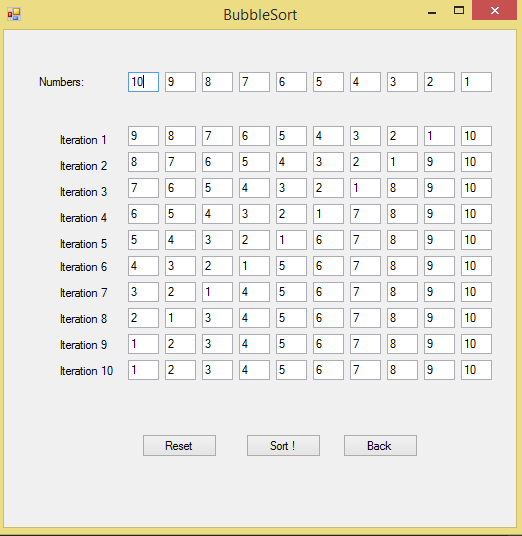
{

Application.Exit();

}

}

}



# MergeSort

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Project\_X\_\_\_Sorting\_Algorithms

{

public partial class MergeSort : Form

{

private int[] arr = new int[10];

List<TextBox> textBoxes = new List<TextBox>();

List<List<TextBox>> lists = new List<List<TextBox>>();

List<TextBox> one = new List<TextBox>();

List<TextBox> two = new List<TextBox>();

List<TextBox> three = new List<TextBox>();

List<TextBox> four = new List<TextBox>();

public MergeSort()

{

InitializeComponent();

textBoxes = new List<TextBox>

{

textBox1,

textBox2,

textBox3,

textBox4,

textBox5,

textBox6,

textBox7,

textBox8,

textBox9,

textBox10

};

lists = new List<List<TextBox>>

{

one,

two,

three,

four,

};

}

private void button1\_Click(object sender, EventArgs e)

{

for (int i = 0; i < 10; i++)

{

try

{

arr[i] = int.Parse(textBoxes[i].Text);

}

catch (Exception)

{

arr[i] = 0;

}

}

merge(arr);

//for (int i = 0; i < 10; i++)

// textBoxes[i].Text = arr[i].ToString();

}

public void merge(int[] array)

{

int y = 0;

if (array.Length < 2)

return;

int step = 1;

int startL, startR;

while (step < array.Length)

{

startL = 0;

startR = step;

while (startR + step <= array.Length)

{

mergeArrays(array, startL, startL + step, startR, startR + step);

startL = startR + step;

startR = startL + step;

}

if (startR < array.Length)

mergeArrays(array, startL, startL + step, startR, array.Length);

step \*= 2;

List<TextBox> gg = lists[y];

for (int x = 0; x < 10; x++)

gg[x].Text = arr[x].ToString();

y++;

}

}

public void mergeArrays(int[] array, int startL, int stopL, int startR, int stopR)

{

int[] right = new int[stopR - startR + 1];

int[] left = new int[stopL - startL + 1];

for (int i = 0, k = startR; i < (right.Length - 1); ++i, ++k)

right[i] = array[k];

for (int i = 0, k = startL; i < (left.Length - 1); ++i, ++k)

left[i] = array[k];

right[right.Length - 1] = int.MaxValue;

left[left.Length - 1] = int.MaxValue;

for (int k = startL, m = 0, n = 0; k < stopR; ++k)

{

if (left[m] <= right[n])

{

array[k] = left[m];

m++;

}

else {

array[k] = right[n];

n++;

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

this.Hide();

Menu M = new Menu();

M.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

for (int k = 0; k < 4; k++)

{

List<TextBox> gg = lists[k];

for (int x = 0; x < 10; x++)

gg[x].Text = "";

}

for (int o = 0; o < 10; o++)

{

textBoxes[o].Text = "";

}

}

private void MergeSort\_FormClosing(object sender, FormClosingEventArgs e)

{

Application.Exit();

}

}

}

