НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



**ЗВІТ**

**Про виконання лабораторної роботи № 1**

**з дисципліни «Алгоритми і структури даних»**

**Лектор:**

Доцент кафедри ПЗ

Коротєєва Т.О.

**Виконав:**

студ. групи ПЗ-23

Михалевич П.-І.В.

**Прийняла:**

Доцент кафедри ПЗ

Мельник Н.Б.

«\_\_\_» \_\_\_\_\_\_ 2021 р.

∑ = \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Львів – 2021

**Тема роботи**: Метод сортування бульбашкою.

**Мета роботи**: Вивчити алгоритм сортування бульбашкою. Здійснити програмну реалізацію алгоритму сортування бульбашкою. Дослідити швидкодію алгоритму сортування бульбашкою.

**Теоретичні відомості**

Сортування бульбашкою— простий алгоритм сортування. Алгоритм працює наступним чином — у поданому наборі даних порівнюються два сусідні елементи. Якщо один з елементів, не відповідає критерію сортування, то ці два елементи міняються місцями. Прохід по списку продовжується до тих пір, доки дані не будуть відсортованими. Оскільки для роботи з елементами масиву він використовує лише порівняння, це сортування на основі порівнянь.

Складність алгоритму рівна О(n²), де n — кількість елементів для сортування. Існує чимало значно ефективніших алгоритмів. Тому даний алгоритм має низьку ефективність у випадках, коли N є досить великим, за винятком, коли заздалегідь відомо, що масив з самого початку буде добре відсотований.

**Покроковий опис алгоритму**

**Алгоритм В.**

Задано масив елементів R1 ,R2 ,…,Rn.

Даний алгоритм реорганізує масив у висхідному порядку, тобто для його елементів буде мати місце співвідношення Ri < Rj - для всіх i,j=1..n.

В1. Цикл за індексом проходження. Повторювати кроки В2 і В3 при i=1..n-1.

В2. Ініціалізація прапорця перестановки: встановити F=0.

В3. Виконання проходження. Повторювати при j=1,2,…,n-i-1 : якщо Rj+1 < Rj , то встановити F=1 та переставити місцями елементи Rj<->Rj+1, ; якщо F=0, то завершити виконання алгоритму.

В4. Кінець. Вихід.

**Постановка завдання**

10. З двох одновимірних масивів цілих чисел сформувати новий, який включає всі парні числа з першого і непарні з другого масиву. Отриманий масив посортувати в порядку зростання.

**Код програми**

**mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QMessageBox>

void **swap**(int \*xp, int \*yp)

{

int temp = \*xp;

\*xp = \*yp;

\*yp = temp;

}

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(*parent*)

, ui(*new* Ui::MainWindow)

{

ui->setupUi(*this*);

connect(ui->pushButton\_clear,SIGNAL(clicked()),*this*,SLOT(MySlot()));

connect(ui->pushButton\_sort,SIGNAL(clicked()),*this*,SLOT(MySlot()));

}

void MainWindow::**MySlot**(){

QPushButton\* btn = (QPushButton\*)sender();

*if*(btn->text() == "Очистити"){

ui->lineEdit\_array\_1->clear();

ui->lineEdit\_array\_2->clear();

ui->textEdit\_steps->clear();

ui->lineEdit\_result->clear();

}

*if*(btn->text() == "Відсортувати"){

QString text = ui->lineEdit\_array\_1->text();

int count\_1 = 0;

int count\_2 = 0;

*if*(!text.isEmpty()){

QStringList valuestr = text.split(',');

count\_1 = valuestr.size();

}*else* QMessageBox::warning(NULL,"Warning!","Enter the array");

text = ui->lineEdit\_array\_2->text();

*if*(!text.isEmpty()){

QStringList valuestr = text.split(',');

count\_2 = valuestr.size();

}*else* QMessageBox::warning(NULL,"Warning!","Enter the array");

int\* array\_1 = *new* int[count\_1]{0};

int\* array\_2 = *new* int[count\_2]{0};

text = ui->lineEdit\_array\_1->text();

*if*(!text.isEmpty()){

QStringList valuestr = text.split(',');

int count = valuestr.size();

*for*(int j = 0; j < count; j++){

*if*(!valuestr[j].isEmpty()){

bool ok;

int value = valuestr[j].toInt(*&ok*);

*if*(ok){

array\_1[j] = value;

}*else* QMessageBox::warning(NULL,"Warning!","Enter the numbers and coma!");

}

}

}

text = ui->lineEdit\_array\_2->text();

*if*(!text.isEmpty()){

QStringList valuestr = text.split(',');

int count = valuestr.size();

*for*(int j = 0; j < count; j++){

*if*(!valuestr[j].isEmpty()){

bool ok;

int value = valuestr[j].toInt(*&ok*);

*if*(ok){

array\_2[j] = value;

}*else* QMessageBox::warning(NULL,"Warning!","Enter the numbers and coma!");

}

}

}

int count\_3 = 0;

*for* (int i = 0; i < count\_1; i++) {

*if*(array\_1[i] % 2 == 0) count\_3++;

}

*for* (int i = 0; i < count\_2; i++) {

*if*(array\_2[i] % 2 == 1) count\_3++;

}

int\* array\_3 = *new* int[count\_3]{0};

int index = 0;

*for* (int i = 0; i < count\_1; i++) {

*if*(array\_1[i] % 2 == 0){

array\_3[index] = array\_1[i];

index++;

}

}

*for* (int i = 0; i < count\_2; i++) {

*if*(array\_2[i] % 2 == 1){

array\_3[index] = array\_2[i];

index++;

}

}

BubbleSort(*array\_3*,count\_3,*ui->textEdit\_steps*);

QString result = "";

*if*(array\_3){

*for* (int i = 0; i < count\_3; i++ ) {

result += QString::number(array\_3[i]);

*if*(i != count\_3-1){

result += ", ";

}

}

}

ui->lineEdit\_result->setText(result);

*delete* [] array\_1;

*delete* [] array\_2;

*delete* [] array\_3;

}

}

void MainWindow::**BubbleSort**(int\* arr,int size, QTextEdit\* textedit){

textedit->setPlainText("");

*for* (int i = 0; i < size-1; i++){

int F = 0;

QString result = "";

*if*(arr){

*for* (int i = 0; i < size; i++ ) {

result += QString::number(arr[i]);

*if*(i != size-1){

result += ", ";

}

}

}

textedit->append(result);

*for* (int j = 0; j < size-i-1; j++) {

*if* (arr[j] > arr[j+1]) {

swap(*&arr[j]*, *&arr[j+*1*]*);

F = 1;

}

}

*if*(F == 0) *break*;

}

}

MainWindow::~***MainWindow***()

{

*delete* ui;

}

**mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QTextEdit>

QT\_BEGIN\_NAMESPACE

*namespace* **Ui** { *class* **MainWindow**; }

QT\_END\_NAMESPACE

*class* **MainWindow** : *public* QMainWindow

{

Q\_OBJECT

*public*:

**MainWindow**(QWidget \*parent = *nullptr*);

~***MainWindow***();

*private* slots:

void **MySlot**();

void **BubbleSort**(int\* array,int size,QTextEdit\* textedit);

*private*:

Ui::MainWindow \*ui;

};

#endif *//* *MAINWINDOW\_H*

**main.cpp**

#include "mainwindow.h"

#include <QApplication>

int main(int argc, char \*argv[])

{

QApplication a(*argc*, *argv*);

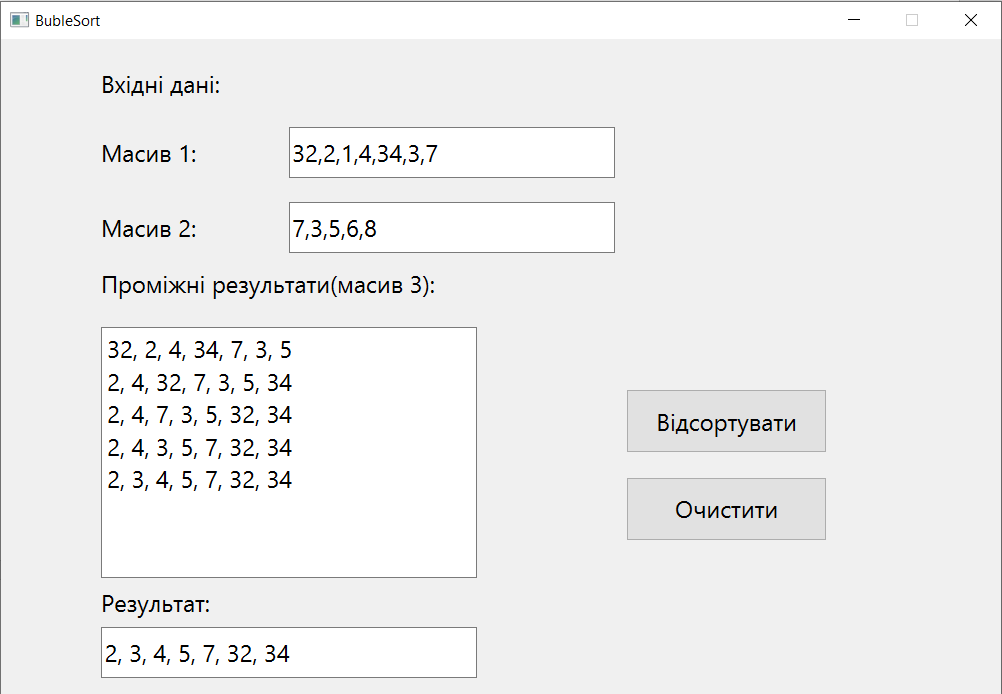
MainWindow w;

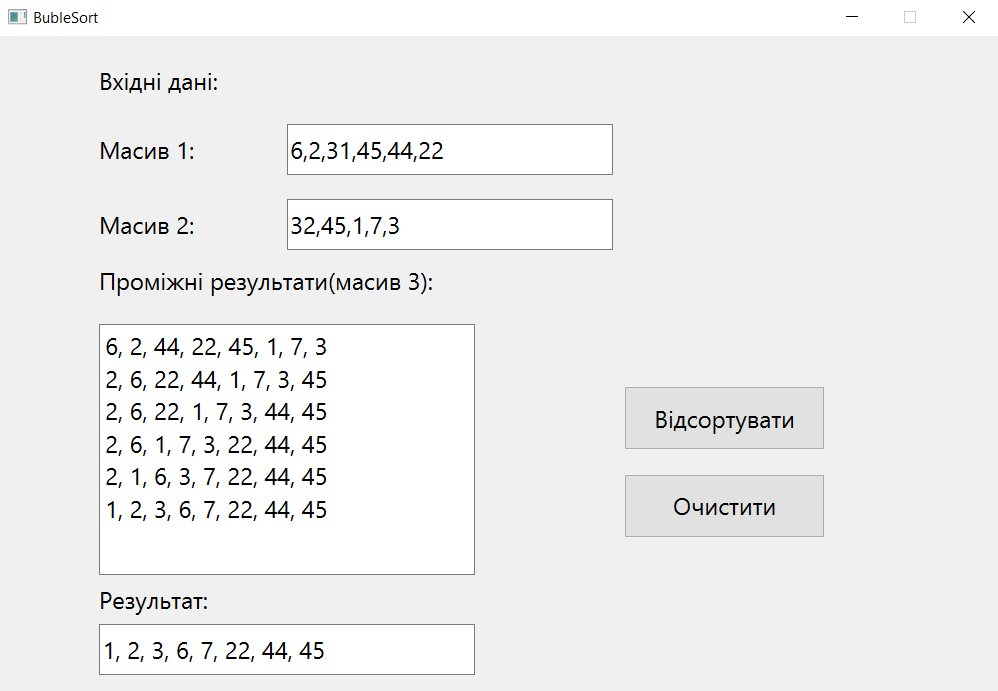
w.show();

*return* a.exec();

}

**Результат програми**





### Висновок

На даній лабораторній роботі я вивчив алгоритм сортування бульбашкою, здійснив програмну реалізацію алгоритму сортування бульбашкою, дослідив швидкодію алгоритму сортування бульбашкою(складність методу О()).