НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



**ЗВІТ**

**Про виконання лабораторної роботи № 2**

**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 простою вибіркою.

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

**Теоретичні відомості**

Сортування простою вибіркою (англійською «Selection Sort») відноситься до алгоритмів сортування на основі вставок. Сенс **сортування вибіркою** полягає у пошуку мінімального значення елемента в масиві, і переміщення цього значення в початок масиву. Ефективність алгоритму наближується до величини n^2, але не досягає її. Алгоритм S має ефективність О(½\*n\*(n-1)).

**Покроковий опис роботи алгоритму**

***Алгоритм PV.***

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.*

PV1. Цикл за індексом проходження. Повторювати кроки S2 - S4 при i=1..n-1.

PV2. Зафіксувати перший поточний елемент: встановити R0 = Ri.

PV3.Пошук найменшого значення min Rj для елементів з індексом j=i+1,i+2,…,n.

PV4. Перестановка елементів. Якщо min Rj < R0 та j !=i , то min Rj <-> R0.

PV5. Кінець. Вихід.

**Постановка завдання**

11. Задано двовимірний масив дійсних чисел. Замінити максимальні елементи кожного рядка на . Впорядкувати (переставити) рядки масиву в порядку зростання їх перших елементів.

**Код програми**

**mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <ctime>

#include <cmath>

#include <QMessageBox>

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\_init,SIGNAL(clicked()),*this*,SLOT(MySlot()));

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

}

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

{

*delete* ui;

}

void MainWindow::**MySlot**()

{

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

*if*(btn->text() == "Clear"){

*for*(int i = 0; i < ui->tableWidget\_enter->rowCount(); i++)

{

*for*(int j = 0; j < ui->tableWidget\_enter->columnCount(); j++)

{

ui->tableWidget\_enter->item(i,j)->setText("");

*delete* ui->tableWidget\_enter->item(i,j);

}

}

*for*(int i = 0; i < ui->tableWidget\_result->rowCount(); i++)

{

*for*(int j = 0; j < ui->tableWidget\_result->columnCount(); j++)

{

ui->tableWidget\_result->item(i,j)->setText("");

*delete* ui->tableWidget\_enter->item(i,j);

}

}

ui->textEdit->clear();

ui->spinBox\_column->*clear*();

ui->spinBox\_column->setValue(ui->spinBox\_column->minimum());

ui->spinBox\_row->*clear*();

ui->spinBox\_row->setValue(ui->spinBox\_row->minimum());

}*else* *if* (btn->text() == "Start"){

ui->textEdit->clear();

int row\_count = ui->tableWidget\_enter->rowCount();

int column\_count = ui->tableWidget\_enter->columnCount();

double\*\* mas = *new* double\*[row\_count];

*for*(int i = 0; i < row\_count; i++)

{

mas[i] = *new* double[column\_count];

}

*//Task*

*for*(int i = 0; i < row\_count; i++)

{

double max = 0;

int maxCol = 0;

*for*(int j = 0; j < column\_count; j++)

{

mas[i][j] = ui->tableWidget\_enter->item(i,j)->text().toDouble();

*if*(mas[i][j] > max){

max = mas[i][j];

maxCol = j;

}

}

mas[i][maxCol] = pow(mas[i][maxCol],1.0/3);

}

*for*(int i = 0; i < row\_count; i++)

{

QString strRow = "";

*for*(int j = 0; j < column\_count; j++)

{

strRow += QString::number(mas[i][j],'f',1) + " ";

}

ui->textEdit->append(strRow);

}

*//* *Selection* *sort*

*for*(int i = 0; i < row\_count-1; i++)

{

int min\_ind = i;

*for* (int k = i+1; k < row\_count; k++) {

*if*(mas[k][0]<mas[min\_ind][0]){

min\_ind = k;

}

}

*if*(min\_ind != i){

*for*(int j = 0; j < column\_count; j++){

double temp = mas[i][j];

mas[i][j] = mas[min\_ind][j];

mas[min\_ind][j] = temp;

}

}

ui->textEdit->append("");

ui->textEdit->append("");

ui->textEdit->append("Step " + QString::number(i+1)+":");

ui->textEdit->append("");

*for*(int i = 0; i < row\_count; i++)

{

QString strRow = "";

*for*(int j = 0; j < column\_count; j++)

{

strRow += QString::number(mas[i][j],'f',1) + " ";

}

ui->textEdit->append(strRow);

}

}

*for*(int i = 0; i < row\_count; i++)

{

*for*(int j = 0; j < column\_count; j++)

{

ui->tableWidget\_result->item(i,j)->setText(QString::number(mas[i][j],'f',1));

}

}

*for*(int i = 0; i < row\_count; i++)

{

*delete* [] mas[i];

}

*delete* [] mas;

}*else* *if* (btn->text() == "Random init"){

srand(time(NULL));

*for*(int i = 0; i < ui->tableWidget\_enter->rowCount(); i++)

{

*for*(int j = 0; j < ui->tableWidget\_enter->columnCount(); j++)

{

QTableWidgetItem\* item = *new* QTableWidgetItem;

double rand\_val = (rand() % 200) / 2.3;

item->setText(QString::number(rand\_val,'f',1));

item->setTextAlignment(Qt::*AlignCenter*);

ui->tableWidget\_enter->setItem(i,j,*item*);

}

}

*for*(int i = 0; i < ui->tableWidget\_result->rowCount(); i++)

{

*for*(int j = 0; j < ui->tableWidget\_result->columnCount(); j++)

{

QTableWidgetItem\* item = *new* QTableWidgetItem;

item->setText("");

item->setTextAlignment(Qt::*AlignCenter*);

ui->tableWidget\_result->setItem(i,j,*item*);

}

}

}

}

void MainWindow::**on\_spinBox\_row\_valueChanged**(int arg1)

{

ui->tableWidget\_enter->setRowCount(arg1);

ui->tableWidget\_result->setRowCount(arg1);

}

void MainWindow::**on\_spinBox\_column\_valueChanged**(int arg1)

{

ui->tableWidget\_enter->setColumnCount(arg1);

ui->tableWidget\_result->setColumnCount(arg1);

}

**mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QPushButton>

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 **on\_spinBox\_row\_valueChanged**(int arg1);

void **on\_spinBox\_column\_valueChanged**(int arg1);

*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**

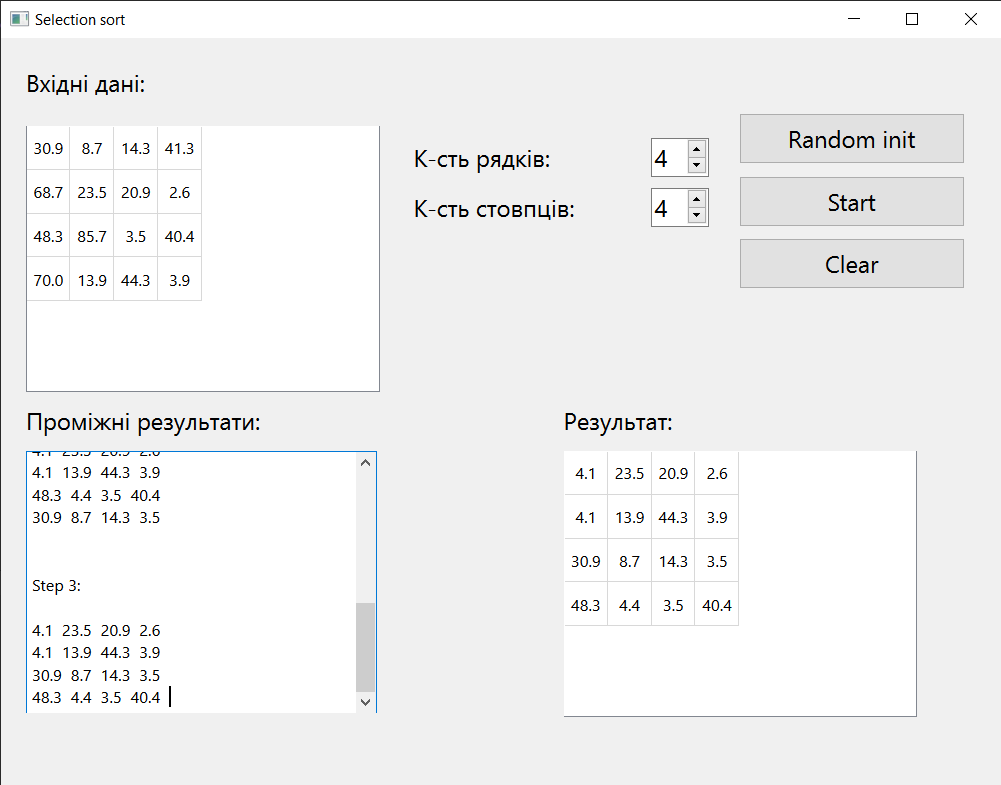


Рис.1

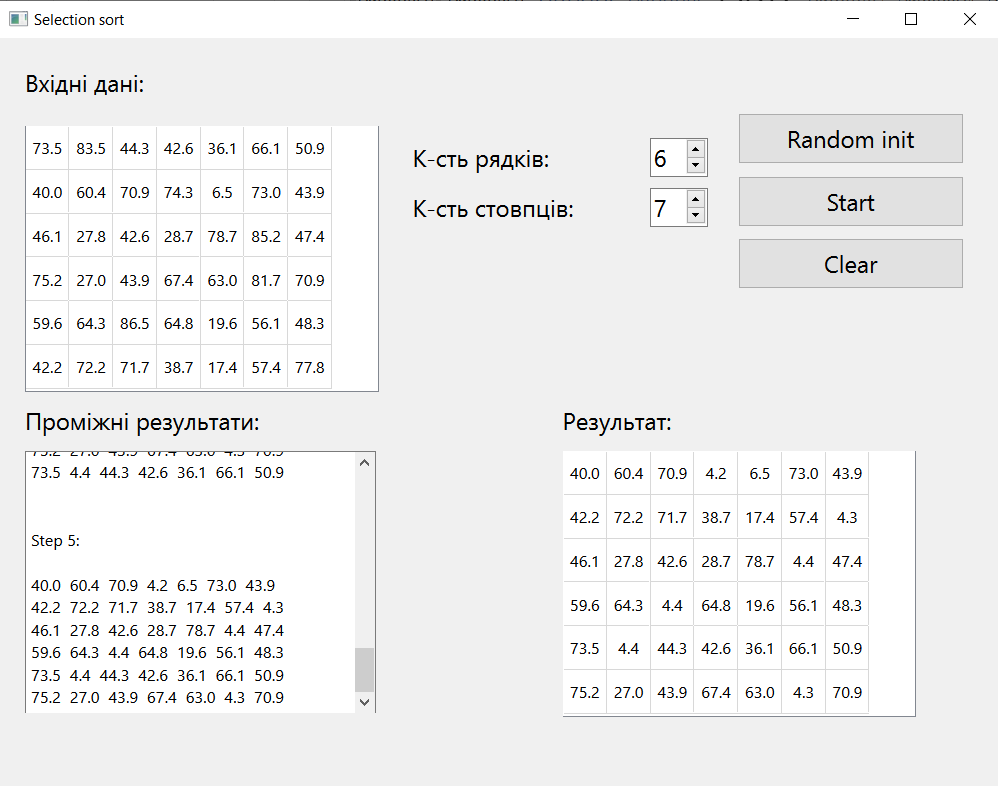


Рис.2

### Висновок

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