­­­­Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення



**Звіт**

Про виконання лабораторної роботи №2

**на тему:**

«Метод сортування вибіркою»

З дисципліни «Алгоритми і структури даних»

**Лектор:**

доц. каф. ПЗ

Коротєєва Т.О.

**Виконав:**

ст. гр. ПЗ-26

Комарницький В.В.

**Прийняв:**

асистент каф. ПЗ

Симець І.І.

« \_\_ » \_\_\_\_\_\_\_\_ 2022 р.

∑ = \_\_\_\_\_ \_\_\_\_\_\_\_\_\_ .

Львів-2022

**Тема роботи:** Метод сортування вибором.

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

**Теоретичний матеріал**

Selection Sort – простий алгоритм сортування лінійного масиву на основі вставок. Має ефективність Тому він не є ефективним для сортування великих масивів.

Алгоритм сортування за вибором сортує масив шляхом повторного знаходження мінімального елемента (враховуючи зростання) з невідсортованої частини та розміщення його на початку. Алгоритм підтримує два підмасиви в заданому масиві:

* Підмасив, який уже відсортовано.
* Залишок підмасиву, який не відсортований.

У кожній ітерації сортування вибору мінімальний елемент (у порядку спадання) з невідсортованого підмасиву вибирається та переміщується до відсортованого підмасиву.

**Алгоритм SS** сортування масиву в якому немає чисел кратних 3 у порядку спадання.

Задано масив елементів

SS1. Цикл за індексом проходження. Повторювати кроки SS2, SS3 і SS4

при і = 0 … n-1.

SS2. Встановлення .

SS3. Виконання проходження. Повторювати при j = i…n;

Якщо .

SS4. Переставити місцями .

SS6. Кінець. Вихід.

**Індивідуальне завдання**

**Варіант 13.**

Задано одномірний масив цілих чисел. Виключити з нього елементи, кратні трьом, а до всіх інших елементів масиву застосувати функцію . Отриманий масив посортувати в порядку спадання.

**Код програми**

void MainWindow::**outputListArray**() {

int sizeArray = 1;

QStringList list = ui->line\_enter->text().split(QLatin1Char(';'), Qt::*SkipEmptyParts*);

sizeArray = list.length();

int ourArray[sizeArray];

QString givenArrayString = "Our array = [ ";

int index = 0;

*for* (int i = 0; i < sizeArray; ++i)

{

ourArray[i] = list[i].toDouble();

*if*(ourArray[i] % 3 != 0){

index += 1;

}

}

*delete* [] array;

size1 = index;

array = *new* int[size1];

int k = 0;

*for*(int i = 0; i < sizeArray; ++i)

{

*for*(int j = k; j < size1; ++j)

{

*if*(ourArray[i] % 3 == 0)

{

*break*;

}

array[j] = ourArray[i] \* ourArray[i];

++k;

*if*(j == 0)

{

givenArrayString += QString::number(array[j]);

*break*;

}

givenArrayString += (", " + QString::number(array[j]));

*break*;

}

}

givenArrayString += " ]";

ui->list\_array->setText(givenArrayString);

}

void MainWindow::**outputRandomList**() {

int numberElements = ui->spinBox\_number->value();

int ourArray[numberElements];

QString givenArrayString = "Our array = [ ";

std::uniform\_int\_distribution<int> distribution (-50000, 50000);

QString initialArray = "[ ";

int index = 0;

*for*(int i = 0; i < numberElements; ++i)

{

ourArray[i] = distribution(\*QRandomGenerator::global());

*if*(ourArray[i] % 3 != 0){

index += 1;

}

*if*(i == 0)

{

initialArray += QString::number(ourArray[i]);

*continue*;

}

initialArray += (", " + QString::number(ourArray[i]));

}

initialArray += " ]";

ui->steps\_list->setText(initialArray);

*delete* [] array;

size1 = index;

array = *new* int[size1];

int k = 0;

*for*(int i = 0; i < numberElements; ++i)

{

*for*(int j = k; j < size1; ++j)

{

*if*(ourArray[i] % 3 == 0)

{

*break*;

}

array[j] = ourArray[i] \* ourArray[i];

++k;

*if*(j == 0)

{

givenArrayString += QString::number(array[j]);

*break*;

}

givenArrayString += (", " + QString::number(array[j]));

*break*;

}

}

givenArrayString += " ]";

ui->list\_array->setText(givenArrayString);

}

void MainWindow::**selectionSort**(int \*array1, int size1) {

int index;

QString steps;

*for* (int i = 0; i < size1-1; ++i)

{

index = i;

*for*(int j = i+1; j < size1; ++j)

{

*if*(array1[j] > array1[index])

{

index = j;

}

}

int tmp = array[i];

array[i] = array[index];

array[index] = tmp;

steps += ("Step " + QString::number(i) + "\n");

*for*(int k = 0; k < size1; ++k)

{

*if*(k == 0)

{

steps += QString::number(array1[k]);

*continue*;

}

steps += (", " + QString::number(array1[k]));

}

steps += "\n";

}

ui->steps\_list->setText(steps);

}

**Протокол виконання програми**

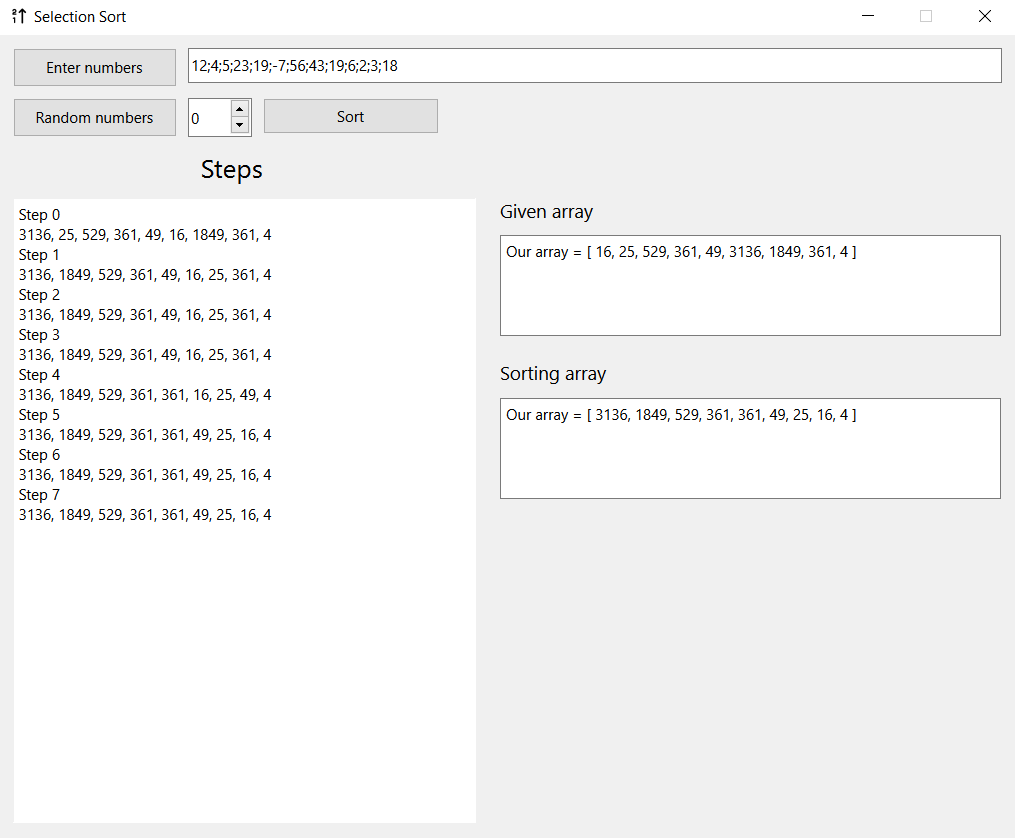


Рис.1. Сортування масиву введеного з клавіатури

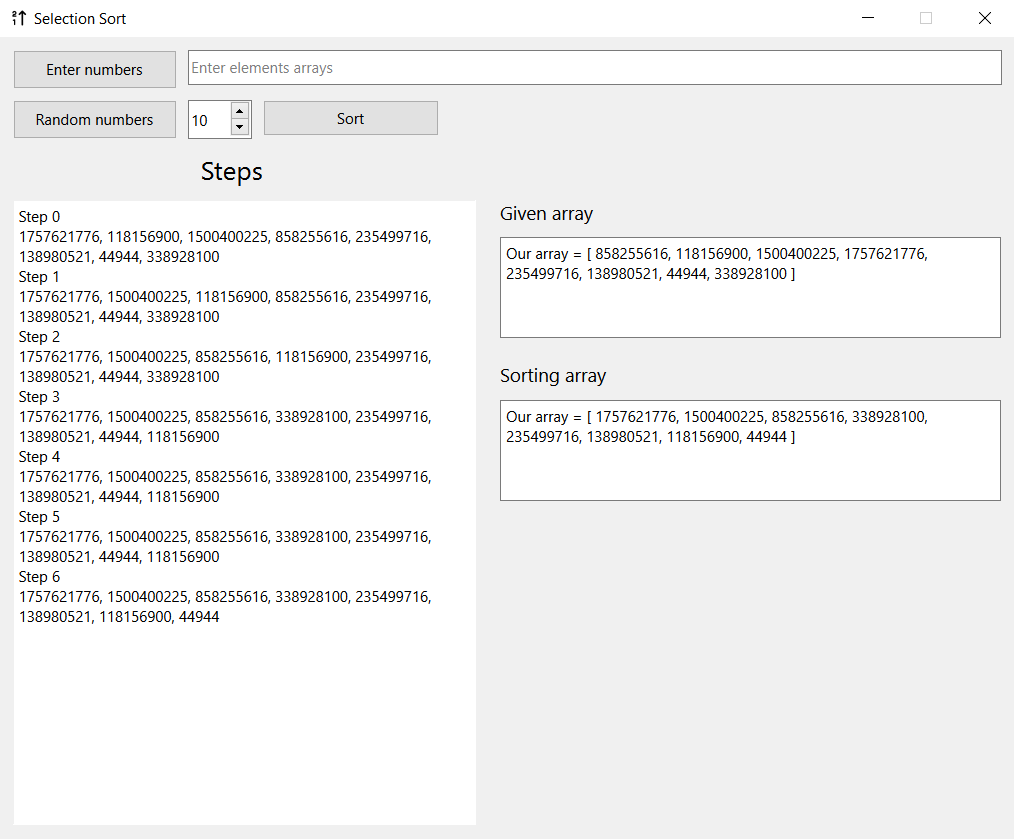


Рис.2. Сортування масиву заповненого випадковими числами

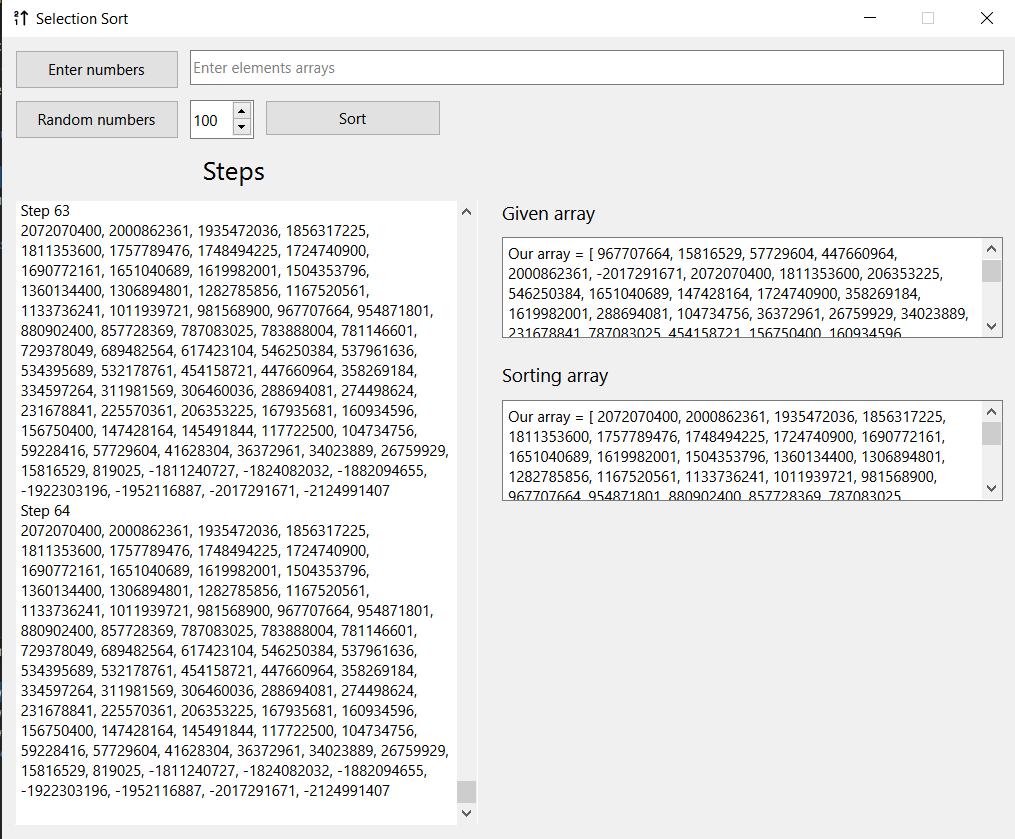


Рис.3. Сортування масиву розміром заповненого випадковими числами

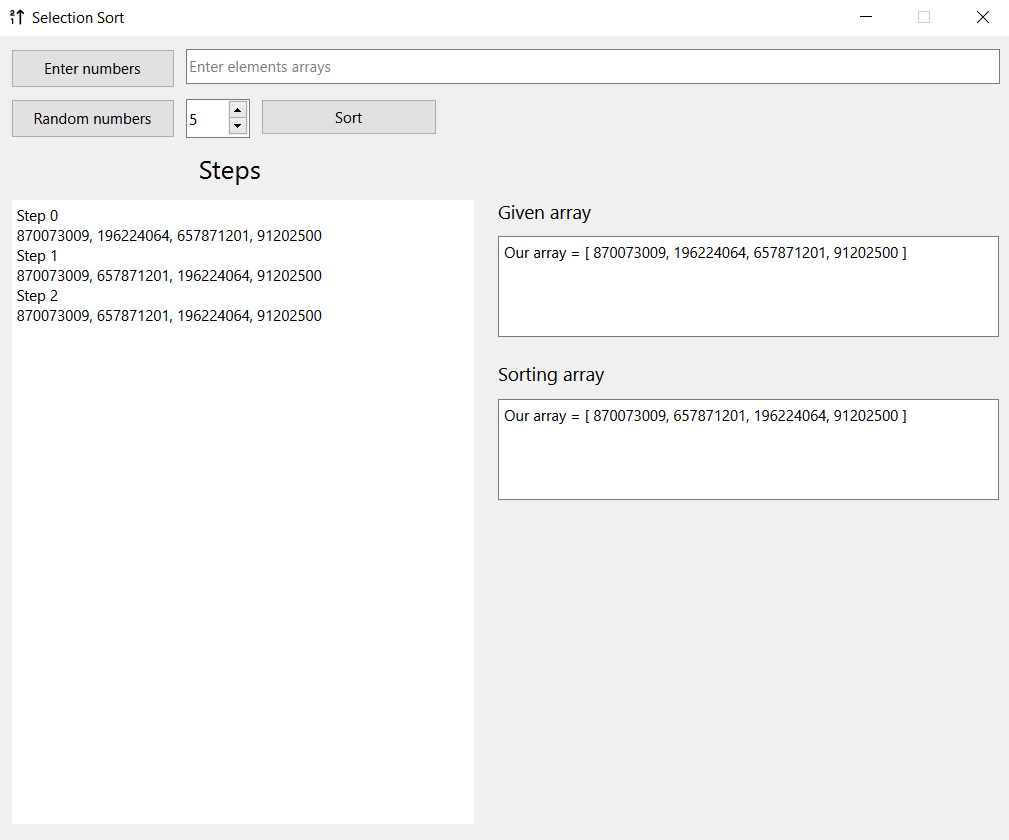


Рис.4. Сортування масиву розміром заповненого випадковими числами

**Висновки**

Під час виконання лабораторної роботи №2 я ознайомився з алгоритмом вибору та реалізував його в програмі з графічним інтерфейсом.