**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

Інститут **ІКНІ**

Кафедра **ПЗ**



**ЗВІТ**

**З дисципліни:** *“Алгоритми і структури даних”*

**На тему:** *“Метод швидкого сортування”*

**Лектор:**

Доцент кафедри ПЗ

Коротєєва Т.О.

**Виконав:**

ст. гр. ПЗ-21

Мартинюк Н.В.

**Прийняв:**

Асистент кафедри ПЗ

Симець І.І.

«\_\_» \_\_\_\_\_\_\_\_\_ 2020 р.

∑ = \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Львів-2020

**Тема:** Метод швидкого сортування.

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

**Завдання для лабораторної роботи:**

1. Ознайомитись з теоретичним матеріалом.

2. В обраному середовищі програмування створити віконний проект та написати програму, яка реалізує алгоритм швидкого сортування.

3. Реалізувати варіант згідно списку підгрупи, додавши 4.

4. Передбачити вивід у вікні вхідних даних, результатів виконання варіанту, проміжних результатів згідно алгоритму та результуючий відсортований масив даних.

5. Оформити звіт про виконання лабораторної роботи.

**Індивідуальне завдання(варіант 8)**

Задано перелік студентів і їх середній бал. Упорядкувати за зростанням середнього балу тих студентів, середній бал яких вищий за 4.

**Теоретичні відомості**

Швидке сортування – це алгоритм сортування, який не потребує додаткової пам’яті і виконує O(n × log(n)) операцій порівнянь. Суть алгоритму в тому, що ми в нашому масиві ви обираємо опорний елемент, потім потрібно поступово відсортувати масив так, щоб зліва від опорного елемента були менші елементи, а справа – більші. Цей алгоритм є рекурсивним.

**Уривок коду програми**

**Індивідуальне завдання**

void QuickSort::IndividualTask()

{

size\_t newSize = 0;

for (size\_t i = 0; i < size; i++)

{

if (students[i].second > 3)

newSize++;

}

pair<string, size\_t>\* tmp = new pair<string, size\_t>[newSize];

size\_t checker = 0;

for (size\_t i = 0; i < size; i++)

{

if (students[i].second < 4)

{

checker++;

continue;

}

tmp[i - checker].first = students[i].first;

tmp[i - checker].second = students[i].second;

}

size = newSize;

delete[]students;

students = tmp;

}

**Алгоритм швидкого сортування**

void QuickSort::QuickSorting(size\_t left, size\_t right)

{

size\_t baseElement = (right + left) / 2;

size\_t i = 0;

size\_t j = right;

if (left < right)

{

if (right - left != 1)

{

while (i < baseElement && j >= baseElement)

{

if (students[i].second >= students[baseElement].second)

{

if (students[j].second < students[baseElement].second)

{

string tmpName = students[i].first;

size\_t tmpAverageMark = students[i].second;

students[i].first = students[j].first;

students[i].second = students[j].second;

students[j].first = tmpName;

students[j].second = tmpAverageMark;

}

else

{

j--;

continue;

}

}

else

{

i++;

continue;

}

}

QuickSorting(left, baseElement - 1);

QuickSorting(baseElement, right);

}

else

{

if (students[left].second > students[right].second)

{

string tmpName = students[left].first;

size\_t tmpAverageMark = students[left].second;

students[left].first = students[right].first;

students[left].second = students[right].second;

students[right].first = tmpName;

students[right].second = tmpAverageMark;

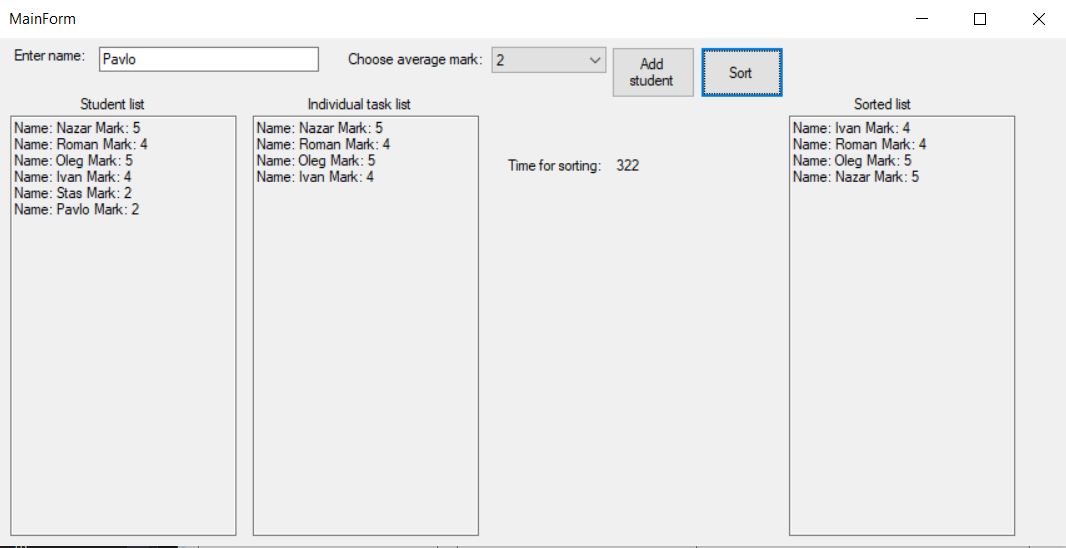
}

}

}

}

**Результат виконання програми**

****

**Висновок**

Виконуючи дану лабораторну роботу, я вивчив алгоритм швидкого сортування, а також закодував програму, яка сортує пару елементів в порядку зростання серeднього балу.