Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій



Звіт

про виконання лабораторної роботи №7.2.2

Опрацювання багатовимірних масивів ітераційним і рекурсивним способами

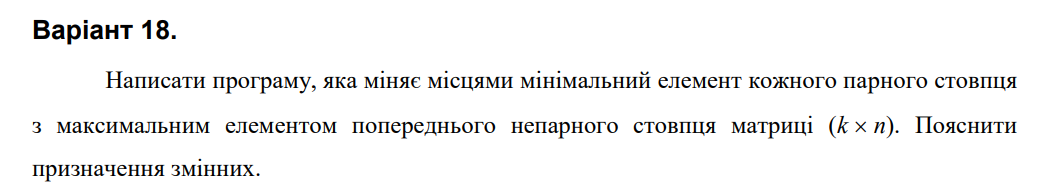
з дисципліни «Алгоритмізація та програмування»

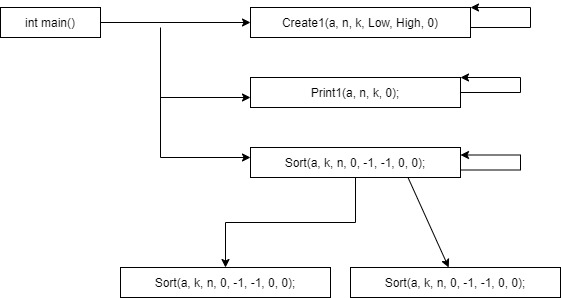
студента IT-11

Полапа Максим

Прийняв доцент Григорович В.Г.

2021

Мета: Навчитися опрацьовувати двовимірні та багатовимірні масиви ітераційним і рекурсивним способами  
Умова: Необхідно: - сформувати матрицю із елементів цілого типу (величину кожної розмірності та діапазон значень елементів – вибрати самостійно); - вивести її на екран у вигляді таблиці, використовуючи форматне виведення; - виконати вказані у завданні дії; - вивести результат, причому, якщо матриця була змінена – то вивести на екран модифіковану матрицю у вигляді окремої таблиці, використовуючи виведення з тими самими специфікаціями формату  
  
  
Структурна схема:



Код програми:  
#include <iostream>

#include <iomanip>

#include <time.h>

using namespace std;

void Create2(int\*\* a, const int n, const int k, const int Low, const int High, int i, int j)

{

if (j < k)

{

a[i][j] = Low + rand() % (High - Low + 1);

return Create2(a, n, k, Low, High, i, j + 1);

}

}

void Create1(int\*\* a, const int n, const int k, const int Low, const int High, int i)

{

if (i < n)

{

Create2(a, n, k, Low, High, i, 0);

return Create1(a, n, k, Low, High, i + 1);

}

}

void Print2(int\*\* a, const int n, const int k, int i, int j)

{

if (j < k)

{

cout << setw(4) << a[i][j];

return Print2(a, n, k, i, j + 1);

}

}

void Print1(int\*\* a, const int n, const int k, int i)

{

cout << endl;

if (i < n)

{

Print2(a, n, k, i, 0);

return Print1(a, n, k, i + 1);

}

cout << endl;

}

int Max(int\*\* a, int n, int i\_max, int a\_max, int i, int j)

{

if (i == 0)

{

a\_max = a[0][j];

}

if (i < n)

{

if (a[i][j] > a\_max)

{

a\_max = a[i][j];

i\_max = i;

return Max(a, n, i\_max, a\_max, i + 1, j);

}

return Max(a, n, i\_max, a\_max, i + 1, j);

}

cout << "max i =" << i\_max << endl;

return i\_max;

}

int Min(int\*\* a, int n, int i\_min, int a\_min, int i, int j)

{

if (i == 0)

{

a\_min = a[0][j];

}

if (i < n)

{

if (a[i][j] < a\_min)

{

a\_min = a[i][j];

i\_min = i;

return Min(a, n, i\_min, a\_min, i + 1, j);

}

return Min(a, n, i\_min, a\_min, i + 1, j);

}

cout << "min i =" << i\_min << endl;

return i\_min;

}

void Sort(int\*\* a, int k, int n, int tmp, int i\_max, int i\_min, int i, int j)

{

if (j < k)

{

if (j % 2 == 0)

{

i\_min = Min(a, n, i\_min, 0, 0, j);

}

else

{

i\_max = Max(a, n, i\_max, 0, 0, j);

}

if (i\_max == -1 || i\_min == -1)

{

return Sort(a, k, n, tmp, i\_max, i\_min, i, j + 1);

}

tmp = a[i\_min][j - 1];

a[i\_min][j - 1] = a[i\_max][j];

a[i\_max][j] = tmp;

i\_max = i\_min = -1;

return Sort(a, k, n, tmp, i\_max, i\_min, i, j + 1);

}

}

int main()

{

srand((unsigned)time(NULL));

int Low = 0;

int High = 20;

int n;

int k;

int min = 0;

int max = 0;

cout << "n = "; cin >> n;

cout << "k = "; cin >> k;

int\*\* a = new int\* [n];

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

{

a[i] = new int[k];

}

Create1(a, n, k, Low, High, 0);

Print1(a, n, k, 0);

Sort(a, k, n, 0, -1, -1, 0, 0);

Print1(a, n, k, 0);

return 0;

}

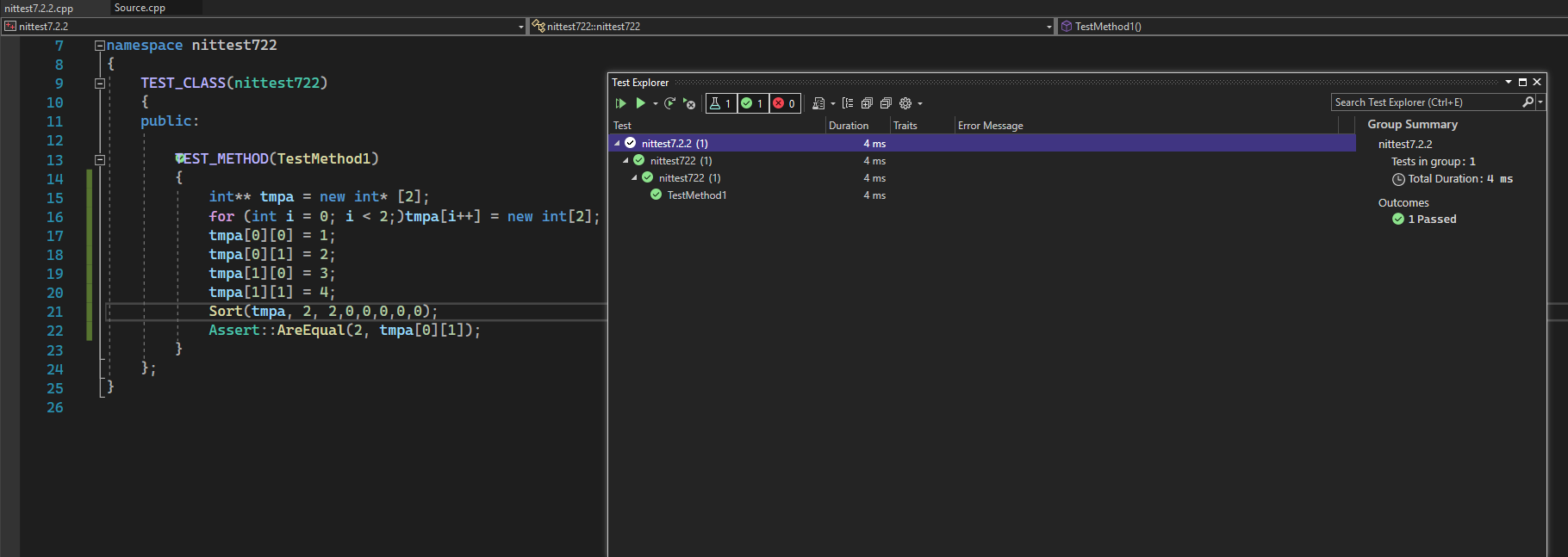
/\*

1,2,4,10

5,6,7,13

8,9,0,9\*/

Git-hub:

Unit-test:  
  
Висновок: На цій лабораторній роботі я навчився опрацьовувати двовимірні та багатовимірні масиви ітераційним і рекурсивним способами