фМІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра прикладної математики та фундаментальних наук

Звіт

до індивідуальної роботи №2

з дисципліни «Паралельні обчислення та засоби їх реалізації»

**Виконав:**

студент групи КНІК-11

Мельніков В. В.

**Прийняв:**

Яджак М.С.

Львів 2022

**Завдання**

Конструкцію (7) реалізувати послідовно на ПК, написавши та відлагодивши для цього відповідну програму на будь-якій мові програмування (C, C++, C#, Java, Python тощо) (Індивідуальне завдання № 2).

**Теоретичні відомості**

*Гіперплощина* – це множина точок n-вимірного простору, яку неможливо зобразити графічно, однак координати цих точок задовольняють лінійне рівняння



Тут a1, a2, ... , an, b – задані коефіцієнти (величини), а x1 , x2 , ... , x­n – змінні. Очевидно, що у трьохвимірному просторі гіперплощина вироджується у *площину*, а в двовимірному – у *пряму лінію*.

*Метод гіперплощин*, як і метод *паралелепіпедів*, може бути використаний для розпаралелювання циклів у векторноконвеєрних, векторнопаралельних та багатопроцесорних обчислювальних системах.

Ідея *методу гіперплощин* розпаралелювання циклів полягає у відшуканні в просторі ітерацій I деякого вихідного циклу такого сімейства паралельних гіперплощин, щоб будь-які дві ітерації p1, p2 , що лежать в одній із цих гіперплощин, були інформаційно і конкуренційно не зв’язані, тобто задовольняли умові:

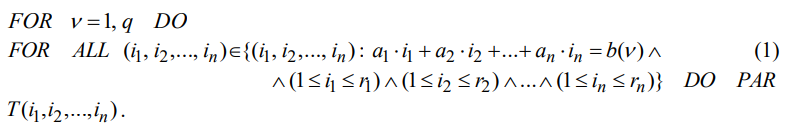


де In(T(p1)) – сукупність вхідних змінних тіла циклу T на ітерації p1;

Out(T(p2)) – сукупність вихідних змінних тіла циклу T на ітерації p2.

За аналогією визначаємо In(T(p2)) та Out(T(p1)). У наведеній вище умові перші дві дужки в об’єднанні означають, що ітерації p1, p2 не зв’язані інформаційно, а третя дужка (тобто її долучення) означає, що згадані ітерації не зв’язані конкуренційно.

Тобто фактично розпаралелювання циклів за методом гіперплощин полягає в перетворенні конструкції із вкладених n циклів з тілом T(i1, i­2, ... ,in) в еквівалентну паралельну конструкцію вигляду:



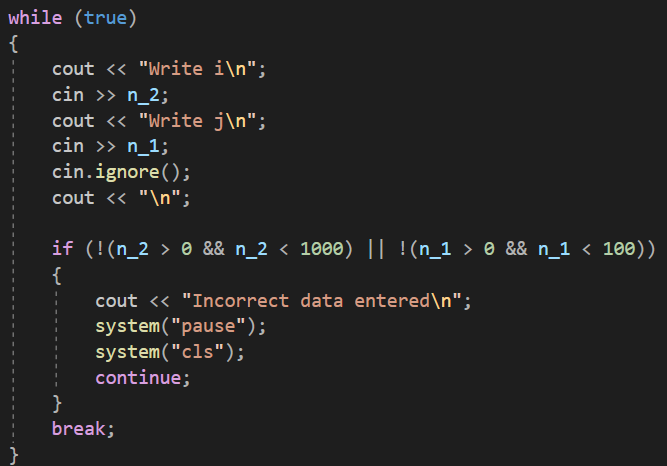
Очевидно, що у наведеній конструкції (1) нам необхідно знайти (обчислити) кількість гіперплощин q і рівняння для цілого їх сімейства.

# Хід роботи

1. Створив відповідний код для вводу кількості ітерацій по “i” та по “j” відповідно

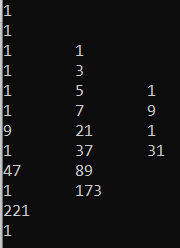


*Рис. 1. Приклад введення I та J для ітерацій*



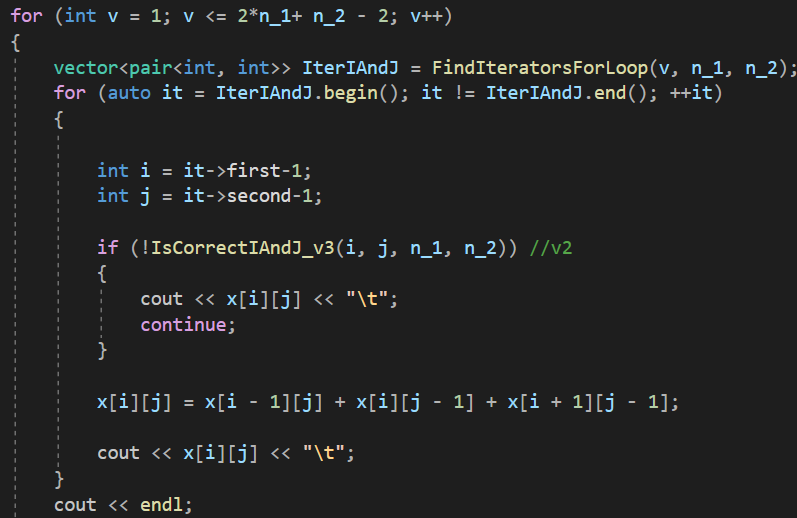
*Рис. 2. Код який відповідає за ввід та перевірку на коректність*

1. Відповідно до конструкції 7 реалізував послідовно цикли на ПК, для прикладу присвоїв всім вхідним даним x одиницю, в результаті отримав вивід:



*Рис. 3. Гіперплощини(14) з відповідними одночасними ітераціями на них*

1. Власне сам код робить перевірки допустимих значень “i” та “j”, таких які будуть належати гіперплощині, в якій зараз проходить ітерація.



*Рис. 4. Код, який реалізує основний принцип даних гіперплощин*

**Додаток 1**

#include <iostream>

#include <vector>

#include <string>

**using** **namespace** std;

vector<pair<**int**, **int**>> FindIteratorsForLoop(**int** v, **int** n\_1, **int** n\_2);

**bool** **IsCorrectIAndJ\_v2**(**int** i, **int** j);

**bool** **IsCorrectIAndJ\_v3**(**int** i, **int** j, **int** n\_1, **int** n\_2);

**int** **main**()

{

**int** n\_2 = **5**;

**int** n\_1 = **7**;

**while** (true)

{

cout << "Write i**\n**";

cin >> n\_2;

cout << "Write j**\n**";

cin >> n\_1;

cin.ignore();

cout << "**\n**";

**if** (!(n\_2 > **0** && n\_2 < **1000**) || !(n\_1 > **0** && n\_1 < **100**))

{

cout << "Incorrect data entered**\n**";

system("pause");

system("cls");

**continue**;

}

**break**;

}

vector <vector<**double**>> x(n\_2, vector<**double**>(n\_1, **1**));

**for** (**int** v = **1**; v <= **2**\*n\_1+ n\_2 - **2**; v++)

{

vector<pair<**int**, **int**>> IterIAndJ = FindIteratorsForLoop(v, n\_1, n\_2);

**for** (**auto** it = IterIAndJ.begin(); it != IterIAndJ.end(); ++it)

{

//x - vector - start from 0

**int** i = it->first-**1**;

**int** j = it->second-**1**;

**if** (!IsCorrectIAndJ\_v3(i, j, n\_1, n\_2)) //v2

{

cout << x[i][j] << "**\t**";

**continue**;

}

x[i][j] = x[i - **1**][j] + x[i][j - **1**] + x[i + **1**][j - **1**];

cout << x[i][j] << "**\t**";

}

cout << endl;

}

}

vector<pair<**int**, **int**>> FindIteratorsForLoop(**int** v, **int** n\_1, **int** n\_2)

{

vector<pair<**int**, **int**>> IterIAndJTmp;

**for** (**int** j = **1**; j <= n\_1; j++)

{

**for** (**int** i = **1**; i <= n\_2; i++)

{

**if** (j - **0.5** \* (-i+v+**2**) == **0**) // j = 0.5\*(-i+v+2)

{

IterIAndJTmp.push\_back(pair<**int**, **int**>(i, j));

}

}

}

**return** IterIAndJTmp;

}

**bool** IsCorrectIAndJ\_v3(**int** i, **int** j, **int** n\_1, **int** n\_2)

{

**if** (i - **1** < **0** || i+**1** >= n\_2 || j - **1** < **0**)

{

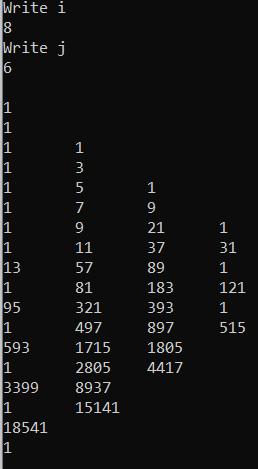
**return** false;

}

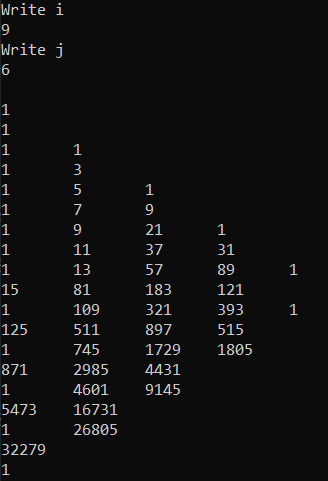
**return** true;

}

**Додаток 2**



*Рис. 1. Вивід при I = 8 та J = 6, всього 18 гіперплощин*



*Рис. 2. Вивід при I = 9 та J = 6 всього 19 гіперплощин*