**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

кафедра ЕОМ



***Лабораторна робота №2***

**з дисципліни “ Паралельні та розподілені обчислення “**

**на тему:**

**“Паралельне представлення алгоритмів”**

Виконав: ст.гр. КІ-34

Алексєєнко Д.С.

Прийняв:

Козак Н.Б.

Львів – 2020

**Мета:** Вивчити можливості паралельного представлення алгоритмів. Набути навиків такого представлення.

**Завдання:**

Запропонувати та реалізувати локально-рекурсивний алгоритм обчислення виразу: , де *А* та *В* матриці з елементами  та , відповідно . Матриця *А* задається однозначно і залежить лише від розмірності даних. Для матриці *В*: заштрихована область — довільні цілі числа, відмінні від нуля, а незаштрихована область - нулі.

№ A B

|  |  |  |
| --- | --- | --- |
| **1** | n 0 .... 0  0 n-1...0  ....  0 .... 1 |  |

**Послідовність виконання роботи:**

Програма з одноразовим присвоюванням.

Програма об’єднана з програмою реалізації оптимізованого локально-рекурсивного алгоритма, і подана в пункті 6.

Рекурсивні рівняння: *Cij(k+1)=Cij(k)+Aij(k)\*Bij(k)*, де *Aij(k)=A[i][j], Bij(k)=B[i][j],* *k* - індекс рекурсії.

Граф залежностей(n=4):



Оптимізований граф залежностей(n=4):



Текст програми, що реалізовує оптимізований локально-рекурсивний алгоритм:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace Pro\_lab2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

int N=0;

int[,] arrayA;

int[,] arrayB;

private void maskedTextBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == Convert.ToChar(13))

{

N = Int32.Parse(maskedTextBox1.Text);

if (N > 13)

{

MessageBox.Show("Матриця завелика для коректного відображення");

return;

}

arrayA = new int[N, N];

arrayB = new int[N, N];

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

{

for (int j = 0; j < N; j++)

{

if (i == j)

{

arrayA[i, j] = (N - i);

}

else

{

arrayA[i, j] = 0;

}

arrayB[i, j] = 0;

}

}

Random rnd = new Random(DateTime.Now.Millisecond);

for (int j = 0; j < N; j++)

{

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

{

arrayB[i, j] = rnd.Next(9)+1;

}

}

Show(arrayA, N, textBox1,true);

Show(arrayB, N, textBox2,false);

}

}

private void Show(int[,] arr,int N, TextBox txt, bool flag)

{

string probil = " ";

string temp = "";

txt.Clear();

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

{

for (int j = 0; j < N; j++)

{

if (i == j && flag)

{

txt.Text += arr[i, j].ToString();

switch (arr[i, j].ToString().Length)

{

case 1:

temp = probil.Remove(0, probil.Length - 4);

txt.Text += temp;

break;

case 2:

temp = probil.Remove(0, probil.Length - 3);

txt.Text += temp;

break;

case 3:

temp = probil.Remove(0, probil.Length - 2);

txt.Text += temp;

break;

case 4:

temp = probil.Remove(0, probil.Length - 1);

txt.Text += temp;

break;

}

}

else

{

txt.Text += arr[i, j].ToString() + probil;

}

}

txt.Text += Environment.NewLine;

}

}

private void button1\_Click(object sender, EventArgs e)

{

Count = 0;

textBox3.Clear();

if (N != Int32.Parse(maskedTextBox1.Text))

{

KeyPressEventArgs e1 = new KeyPressEventArgs(Convert.ToChar(13));

maskedTextBox1\_KeyPress(sender, e1);

}

ArrayMul(arrayA, arrayB, N);

localAlgorutm(arrayA, arrayB, N);

}

int Count = 0;

void ArrayMul(int[,] arrA, int[,] arrB, int N)

{

int[,,] arrC = new int[N, N, N+1];

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

{

for (int j = 0; j < N; j++)

{

for (int k = 0; k < N; k++)

{

arrC[i, j, k + 1] = arrC[i, j, k] + arrA[i, k] \* arrB[k, j];

Count += 2;

}

}

}

Show(arrC, N,"З одноразовим присвоєнням", Count);

}

private void localAlgorutm(int[,] arrA, int[,] arrB, int N)

{

int q = 0;

int[,,] arrC = new int[N, N, N+1];

for (int j = 0; j < N; j++)

{

for (int i = 0, k = 0; i <= j && k <N ; i++,k++)

{

arrC[i, j, N] = arrA[i, i] \* arrB[k, j];

q++;

}

}

Show(arrC, N,"Локально-рекурсивний алгоритм" ,q);

}

private void Show(int[,,] arrC, int N, string Mess, int q)

{

textBox3.Text += Mess + Environment.NewLine;

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

{

for (int j = 0; j < N; j++)

{

textBox3.Text += arrC[i, j, N].ToString() + "\t";

}

textBox3.Text += Environment.NewLine;

}

textBox3.Text += "----------------------------------------------------" + Environment.NewLine;

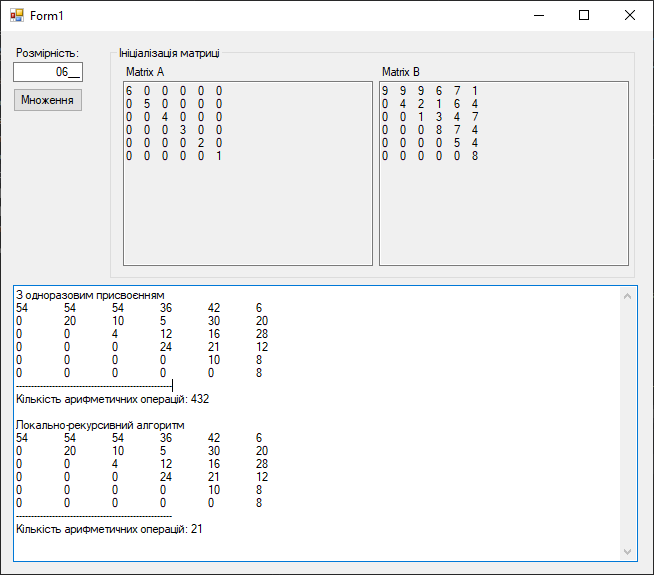
textBox3.Text += "Кiлькiсть арифметичних операцій: " + q.ToString() + Environment.NewLine;

}

}

}

Результат виконання програми для n = 6;



Висновок: на цій лабораторній роботі я вивчив можливості паралельного представлення алгоритмів. Виходячи з оптимізованого графу залежностей ми отримали високу ефективність обчислень.