**Кафедра телекомунікацій**

**Лабораторна робота №4**

з дисципліни

«Дискретна математика»

Виконав: студент групи ІК-11

Снігур Стефан

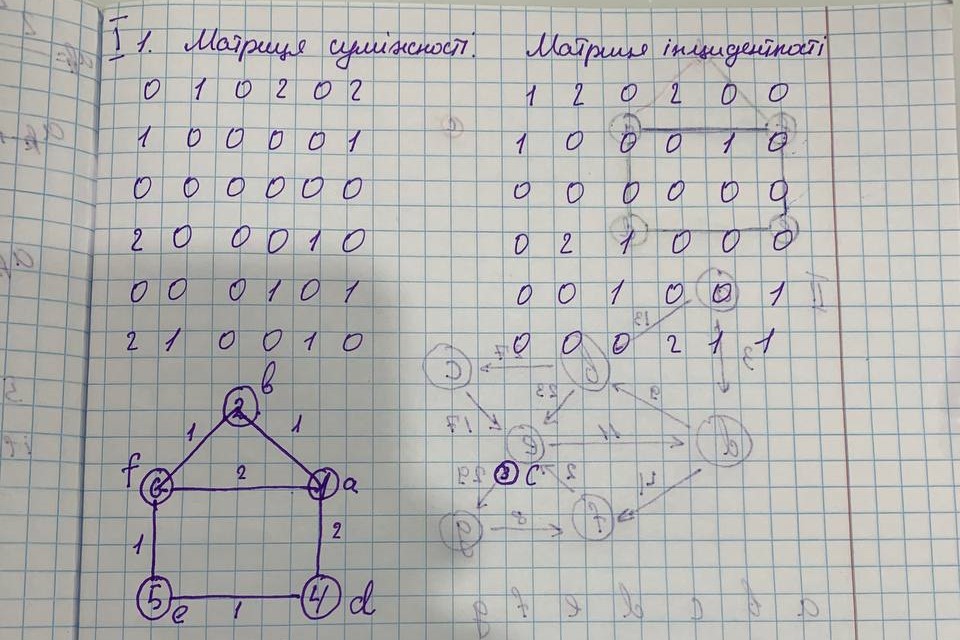
Викладач: Юринець Р.В.

Львів – 2022 р.

***Завдання 1.***

Побудувати матрицю суміжності та матрицю інцидентності для заданого графа. Намалювати граф.





***Завдання 2.***

*Визначити тип графа.*

Граф простий

***Завдання 3.***

Виписати усі ейлерові та гамільтонові ланцюги та цикли.

а) Граф не має Єйлерового циклу

б) Ейлеровий ланцюг : 6→1→2→6→5→4→1

в) Граф не має Гамільтонового циклу

г) Граф не має Гамільтонового ланцюга

***Завдання 4.***

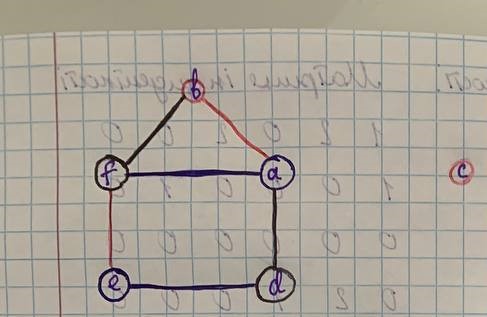
Визначити хроматичне та реберне хроматичне число графа.

Хроматичне число = 3.

Реберне хроматичне число = 3.

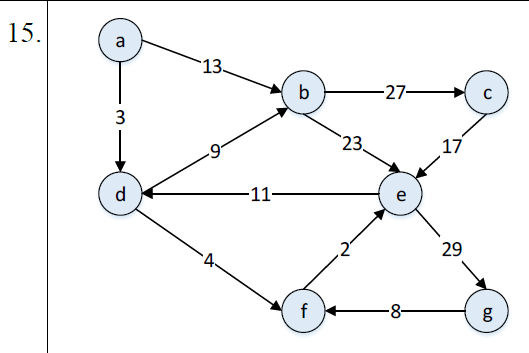
***Завдання 5.***

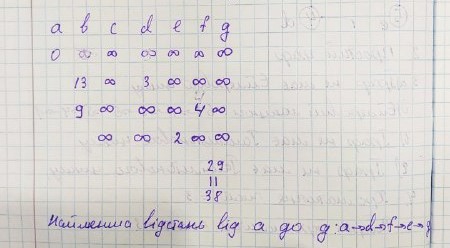
Розфарбувати вершини та ребра графа.



***Завдання 6.***

За алгоритмом Дейкстри знайти найкоротшi вiддалi вiд вершини *a* до вершини *g* графа з табл. 2.





Програма побудови матриці суміжності простого графа.

#include <stdio.h>

#include <conio.h>

#include <iostream>

#include <math.h>

using namespace std;

void main()

{

int x, y, w, chromatic;

int summond[6];

int m[15] = { 1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4, 5 };

int n[15] = { 2, 3, 4, 5, 6, 3, 4, 5, 6, 4, 5, 6, 5, 6, 6 };

int contiguity[6][6];

int incidence[6][11];

int Rmn[15];

cout << "Input values:\n";

for (int i = 0; i <= 14; i++)

{

cin >> Rmn[i];

}

int k = 0, j = 0, z = -1;

for (int i = 0; i <= 5; i++)

{

z += 1;

for (j = z; j <= 5; j++)

{

if (i == j)

{

contiguity[i][j] = 0;

}

else

{

contiguity[i][j] = Rmn[k];

contiguity[j][i] = Rmn[k++];

}

}

}

cout << "\nMatrix of contiguity:\n\n";

for (int i = 0; i <= 5; i++)

for (int k = 0, j = 0; j <= 5; j++)

{

cout << contiguity[i][j] << " ";

if (j == 5) cout << "\n";

}

for (int i = 0; i <= 5; i++)

for (int j = 0; j <= 10; j++)

incidence[i][j] = 0;

for (int i = 0, k = 0; i <= 14; i++)

{

x = Rmn[i];

y = m[i] - 1;

w = n[i] - 1;

while (x > 0)

{

incidence[y][k] = 1;

incidence[w][k] = 1;

k++;

x--;

}

}

\_getch();

}

Програма перетворення матриці суміжності в матрицю інцидентності

#include <stdio.h>

#include <Windows.h>

#include <conio.h>

int main()

{

int i, j, v, n, m, k;

char a[100][100];

char\* in[100];

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

printf("Введіть розмірність матриці n =");

scanf\_s("%d", &n);

printf("Введіть розмірність матриці m =");

scanf\_s("%d", &m);

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

for (j = 0; j < n; j++)

{

printf("a[%d][%d]= ", i + 1, j + 1);

scanf\_s("%d", &a[i][j]);

}

printf("Матриця сміжності: \n");

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

{

for (j = 0; j < n; j++)

printf("|%d| ", a[i][j]);

printf("\n");

}

for (i = v = 0; i < n; i++) for (j = i + 1; j < m; j++) if (a[i][j]) v++;

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

in[i] = (char\*)

malloc(v); memset(in[i], 0, n);

}

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

for (j = i + 1; j < m; j++) if (a[i][j]) {

in[k][i] = in[k][j] = 1;

k++;

}

printf("Матриця інцидентностей : \n");

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

{

for (j = 0; j < m; j++)

printf("|%d| ", a[i][j]);

printf("\n");

}

system("pause");

return 0;

}

Висновок: виконавши цю лабораторну роботу я вивчив основні властивості графів, способи подання графів, шляхи та цикли, обходу графі, розфарбовування графів, набув практичних навичок програмування алгоритмів, що базуються на графах.