**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**



**Лабораторна робота №4**

з дисципліни

«Дискретна математика»

**Виконав:**

студент групи КН-110

Король Орест

**Викладач:**

Мочурад Л.І.

Львів – 2017 рік

*Лабораторна робота № 4*

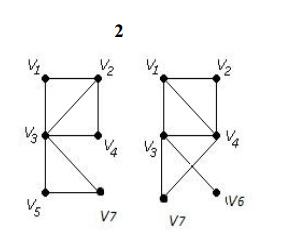
***Тема роботи:***Основні операції над графами. Знаходження остова

мінімальної ваги за алгоритмом Прима-Краскала.

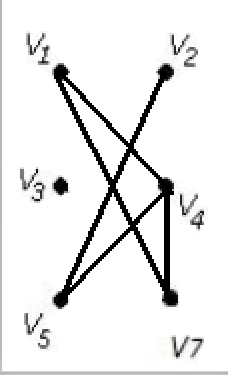
***Мета роботи:*** набуття практичних вмінь та навичок з використання

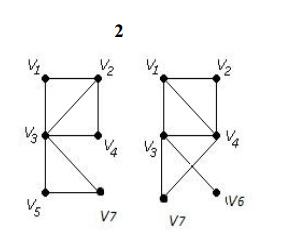
алгоритмів Прима і Краскала.

**Варіант 7**

**Завдання № 1.** Розв'язати на графах наступні задачі:  
**1.1)** знайти доповнення до першого графу,  
**1.2)** об’єднання графів,  
**1.3)** кільцеву суму G1 та G2 (G1+G2),  
**1.4)** розщепити вершину у другому графі,  
**1.5)** виділити підграф А, що складається з 3-х вершин в G1 і знайти  
стягнення А в G1 (G1\ A),

**1.6)** добуток графів.

**Розв’язок:**

**1)**

**2)**

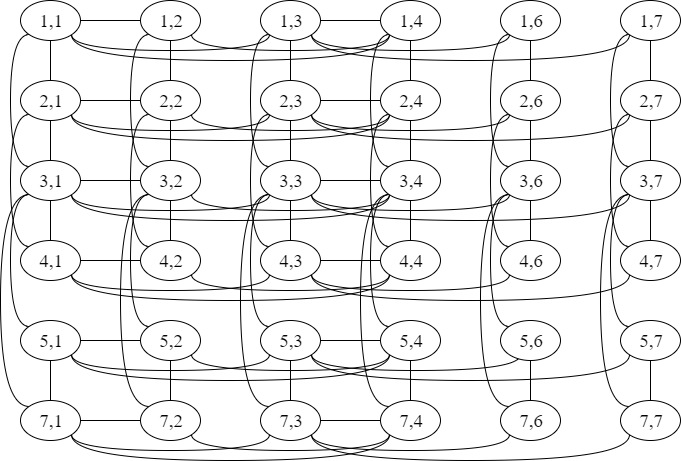
**3)**

**4)**  **–** розщеплена вершина( і )

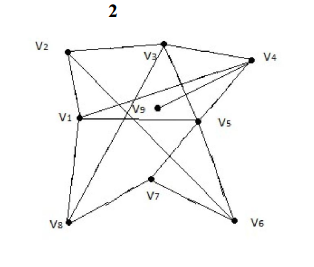
**5)** Підграф А, що складається з трьох вершин

Стягнення і

**6)**

****

2. Знайти таблицю суміжності та діаметр графа.



Розв’язок:

Таблиця суміжності

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
|  | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
|  | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
|  | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
|  | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
|  | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
|  | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

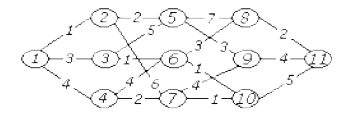
Діаметр графа = 3

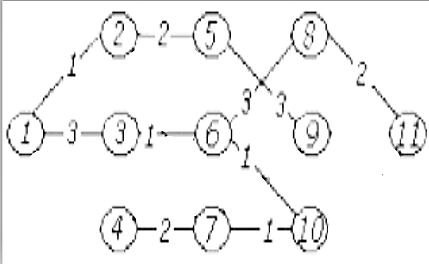
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  | 0 | 1 | 2 | 1 | 1 | 2 | 2 | 1 | 2 |
|  | 1 | 0 | 1 | 2 | 2 | 1 | 2 | 2 | 3 |
|  | 2 | 1 | 0 | 1 | 1 | 1 | 2 | 1 | 2 |
|  | 1 | 2 | 1 | 0 | 1 | 2 | 2 | 2 | 1 |
|  | 1 | 2 | 1 | 1 | 0 | 1 | 1 | 2 | 2 |
|  | 2 | 1 | 2 | 2 | 1 | 0 | 1 | 2 | 3 |
|  | 2 | 2 | 2 | 1 | 1 | 1 | 0 | 1 | 3 |
|  | 1 | 2 | 1 | 2 | 2 | 2 | 1 | 0 | 3 |
|  | 2 | 3 | 2 | 1 | 2 | 3 | 3 | 3 | 0 |

**3.** Знайти двома методами (Краскала і Прима) мінімальне остове  
дерево графа.

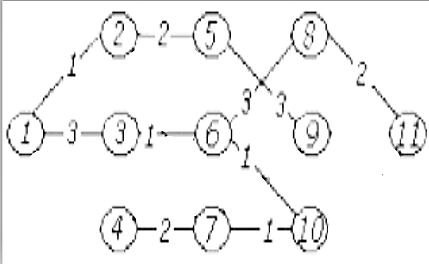
**Розв’язок:**

Метод Прима (початкова вершина 1)





Метод Краскала



**Завдання №2.** Написати програму, яка реалізує алгоритм  
знаходження остового дерева мінімальної ваги згідно свого варіанту.

**Хід роботи**

**Текст програми**

#include <iostream>

using namespace std;

int main(void)

{

int graf[11][11] = { { 0,7,3,1,0,0,0,0,0,0,0 },{ 7,0,0,0,2,0,1,0,0,0,0 },{ 3,0,0,0,7,4,0,0,0,0,0 },{ 1,0,0,0,0,2,5,0,0,0,0 },{ 0,2,7,0,0,0,0,4,4,0,0 },{ 0,0,4,2,0,0,0,6,0,2,0 },{ 0,1,0,5,0,0,0,0,3,3,0 },{ 0,0,0,0,4,6,0,0,0,0,7 },{ 0,0,0,0,4,0,3,0,0,0,4 },{ 0,0,0,0,0,2,3,0,0,0,5 },{ 0,0,0,0,0,0,0,7,4,5,0 } };

int new\_graf[11][11] = { { 0,0,0,0,0,0,0,0,0,0,0 },{ 0,0,0,0,0,0,0,0,0,0,0 },{ 0,0,0,0,0,0,0,0,0,0,0 },{ 0,0,0,0,0,0,0,0,0,0,0 },{ 0,0,0,0,0,0,0,0,0,0,0 },{ 0,0,0,0,0,0,0,0,0,0,0 },{ 0,0,0,0,0,0,0,0,0,0,0 },{ 0,0,0,0,0,0,0,0,0,0,0 },{ 0,0,0,0,0,0,0,0,0,0,0 },{ 0,0,0,0,0,0,0,0,0,0,0 },{ 0,0,0,0,0,0,0,0,0,0,0 } };

int findI, findJ, findNum, h;

bool check[11] = {false,false,false, false, false, false, false, false, false, false, false};

bool chp;

cin >> h;

cout << "Primal graph matrix:" << endl;

for (int i = 0; i < 11; i++){

for (int j = 0; j < 11; j++){

cout << graf[i][j] << " "; }

cout << endl; }

findI = 0;

findJ = 0;

findNum = 100;

for (int i = 0; i < 11; i++) {

for (int j = 0; j < 11; j++) {

if (graf[i][j]<findNum && graf[i][j]!=0) {

findNum = graf[i][j]; }}}

chp = true;

for (int i = 0; i < 11; i++){

for (int j = 0; j < 11; j++){

if (graf[i][j] == findNum && chp == true) {

findI = i;

findJ = j;

check[i] = true;

check[j] = true;

new\_graf[i][j] = graf[i][j];

chp = false;}

}}

for (int i = 0; i < 11; i++) {

if (check[i] == true) {

for (int j = 0; j < 11; j++) {

graf[i][j] = 0; }}}

for (int q = 0; q < 9; q++) {

findI = 0;

findJ = 0;

findNum = 100;

for (int i = 0; i < 11; i++) {

for (int j = 0; j < 11; j++) {

if (check[j] == true && graf[i][j]<findNum && graf[i][j] != 0) {

findNum = graf[i][j]; }}}

chp = true;

for (int i = 0; i < 11; i++) {

for (int j = 0; j < 11; j++) {

if (check[j] == true && graf[i][j] == findNum && chp == true) {

findI = i;

findJ = j;

check[i] = true;

check[j] = true;

new\_graf[i][j] = graf[i][j];

chp = false;

}}}

for (int i = 0; i < 11; i++) {

if (check[i] == true) {

for (int j = 0; j < 11; j++) {

graf[i][j] = 0;

}

}}

}

cout << endl << endl << "Tree matrix:" << endl;;

for (int i = 0; i < 11; i++) {

for (int j = 0; j < 11; j++) {

cout << new\_graf[i][j] << " "; }

cout << endl; }

for (h; h < 11; h++) {

for (int j = 0; j < 11; j++) {

if (new\_graf[h][j]>0) {

cout << h+1 << "--" << j+1 << endl;

if (new\_graf[h==11][j]>0){

h=0;

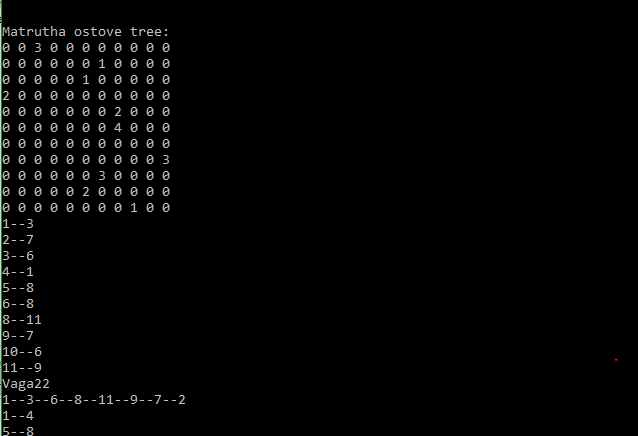
cout << h+1 << "--" << j+1 << endl;

}

}}

}}

**Приклад виконання програми**



***Висновок:*** Я набув практичних вмінь та навичок з використання

алгоритмів Пріма і Краскала.