

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

Розрахункова робота

з дисципліни

«Дискретна математика»

Виконав:

студент групи КН-115

Чеханівський Михайло

Викладач:

Мельникова Н.І.

Львів – 2019 р.

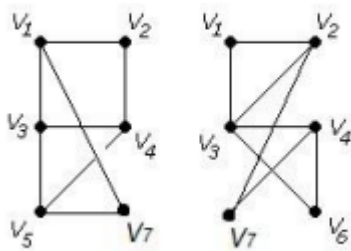
Варіант №10

Завдання № 1

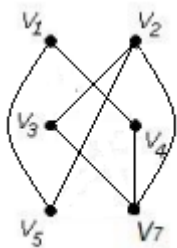
Виконати наступні операції над графами:

- 1) знайти доповнення до першого графу
- 2) об'єднання графів
- 3) кільцеву сумму $G1$ та $G2$ ($G1+G2$)
- 4) розмножити вершину у другому графі
- 5) виділити підграф A - що складається з 3-х вершин в $G1$
- 6) добуток графів.

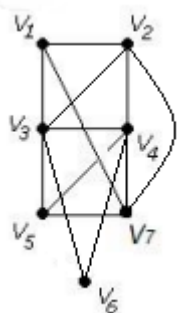
10)



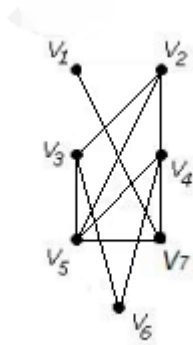
1)



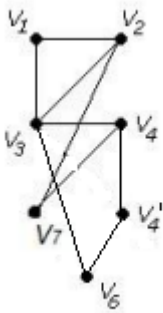
2)



3)



4)

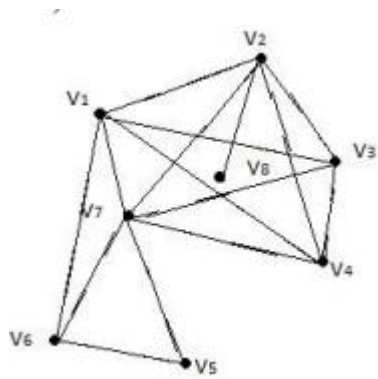


5)



Завдання № 2

Скласти таблицю суміжності для орграфа.



	V1	V2	V3	V4	V5	V6	V7	V8
V1	0	1	1	1	0	1	1	0
V2	1	0	1	1	0	0	1	1
V3	1	1	0	1	0	0	1	0
V4	1	1	1	0	0	0	1	0
V5	0	0	0	0	0	1	1	0
V6	1	0	0	0	1	0	1	0
V7	1	1	1	1	1	1	0	0
V8	0	1	0	0	0	0	0	0

Завдання № 3

Для графа з другого завдання знайти діаметр.

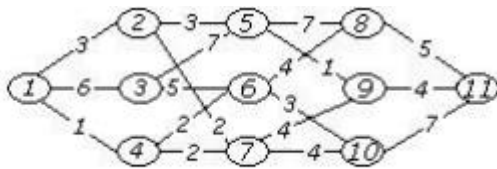
	V1	V2	V3	V4	V5	V6	V7	V8
V1	0	1	1	1	2	1	1	2
V2	1	0	1	1	2	2	1	1
V3	1	1	0	1	2	2	1	2
V4	1	1	1	0	2	2	1	2
V5	2	3	3	2	0	1	1	3
V6	1	2	2	2	1	0	1	3
V7	1	1	1	1	1	1	0	2
V8	2	1	2	2	3	3	2	0

Отже 3 - діаметр графа.

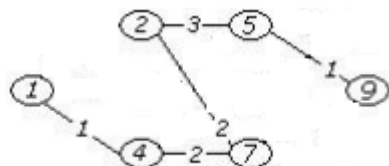
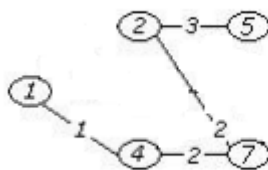
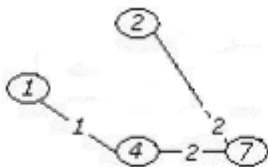
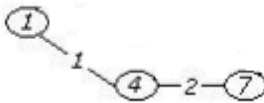
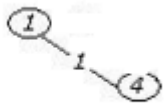
Завдання № 5

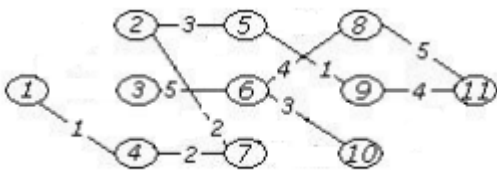
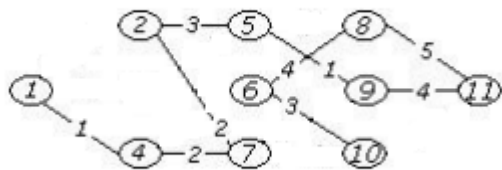
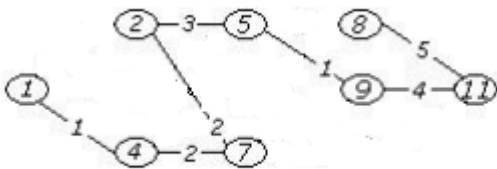
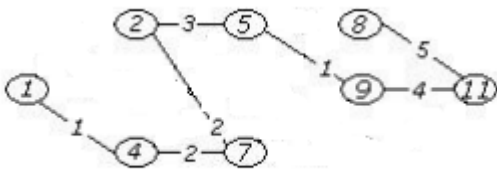
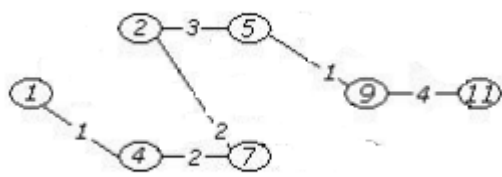
Знайти двома методами (Краскала і Прима) мінімальне остове дерево графа.

10)

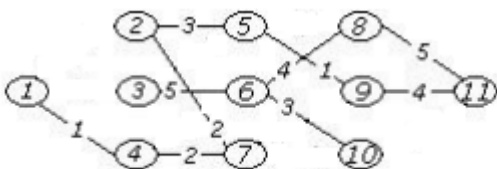


Знаходження мінімального остового дерева методом Прима:



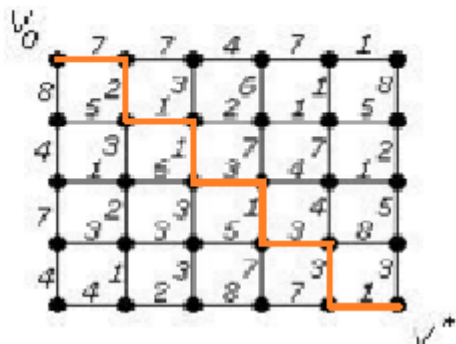


Знаходження мінімального остового дерева методом Краскала:



За допомогою алгоритму Дейкстри знайти найкоротший шлях у графі між парою вершин V_0 і V^* .

10)



Найкоротший шлях $V_1, V_7, V_8, V_{14}, V_{15}, V_{21}, V_{22}, V_{28}$

Довжина=22

```
#include <iostream>
using namespace std;
const int n = 9;
int i, j;
bool visited[9];

int graph[n][n] = { {0,1,0,0,0,0,1,1,1},
{1,0,1,0,0,0,0,1,0},
{0,1,0,1,0,1,0,1,0},
{0,0,1,0,1,0,1,0,0},
{0,0,0,1,0,1,0,1,0},
{0,0,1,0,1,0,0,1,0},
{1,0,0,1,0,0,0,1,0},
{1,1,1,0,1,1,1,0,0},
{1,0,0,0,0,0,0,0,0}
};

void DFS(int st){
    int r;
    cout << st + 1 << " ";
    visited[st] = true;
    for (r = 0; r < n; r++)
        if ((graph[st][r] != 0) && (!visited[r]))
            DFS(r);
}
```

```

void main(){
    int start;
    cout << "Matrix: " << endl;
    for (i = 0; i < n; i++){
        visited[i] = false;
        for (j = 0; j < n; j++)
            cout << " " << graph[i][j];
        cout << endl;
    }
    cout << "Start point "; cin >> start; //прав 9 за стартову
    cout << "Stack: ";
    DFS(start - 1);
}

```

```

Matrix:
0 1 0 0 0 0 1 1 1
1 0 1 0 0 0 0 1 0
0 1 0 1 0 1 0 1 0
0 0 1 0 1 0 1 0 0
0 0 0 1 0 1 0 1 0
0 0 1 0 1 0 0 1 0
1 0 0 1 0 0 0 1 0
1 1 1 0 1 1 1 0 0
1 0 0 0 0 0 0 0 0
Start point 9
Stack: 9 1 2 3 4 5 6 8 7

```