

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА
ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

Лабораторна робота
з дисципліни
«Дискретна математика»

Виконав:
студент групи КН-109
Кіндрат
Володимир
Викладач:
Мельникова Н.І.

Львів – 2018 р.

Лабораторна робота № 4.

Тема: Основні операції над графами. Знаходження остова мінімальної ваги за алгоритмом Пріма-Краскала

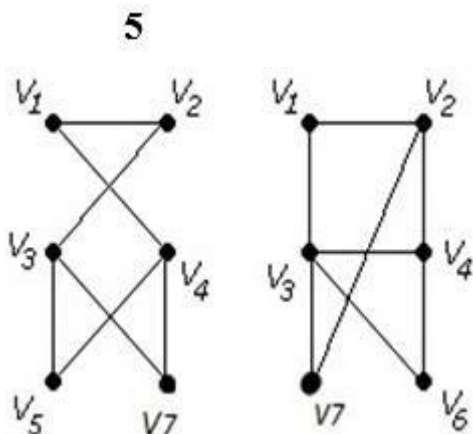
Мета роботи: набуття практичних вмінь та навичок з використання алгоритмів Пріма і Краскала.

Варіант № 5

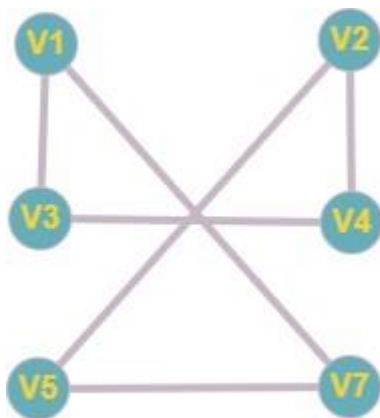
Завдання № 1. Розв'язати на графах наступні задачі:

Виконати наступні операції над графами:

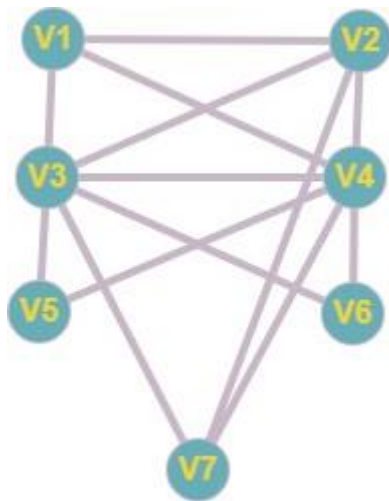
- 1) знайти доповнення до першого графу,
- 2) об'єднання графів,
- 3) кільцеву суму $G1$ та $G2$ ($G1+G2$),
- 4) розщепити вершину у другому графі,
- 5) виділити підграф A , що складається з 3-х вершин в $G1$ і знайти стягнення A в $G1$ ($G1 \setminus A$)
- 6) добуток графів.



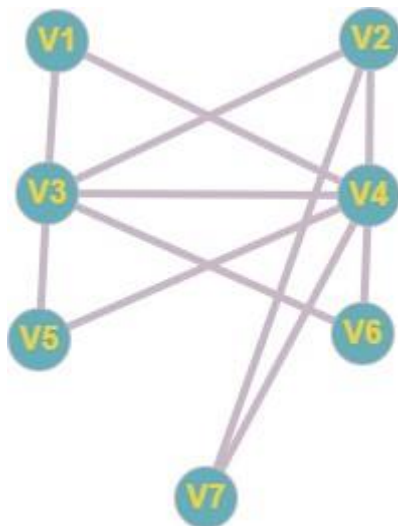
- 1) Доповнення до першого графу:



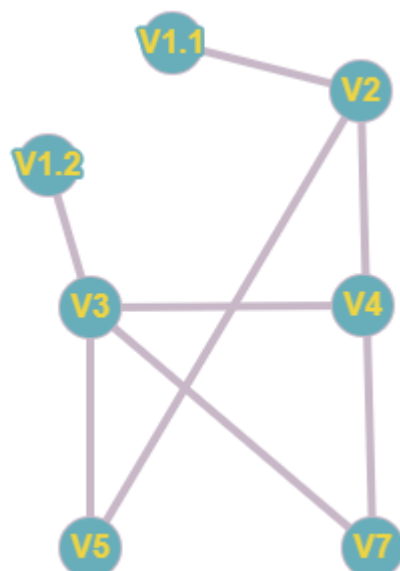
2) Об'єднання графів:



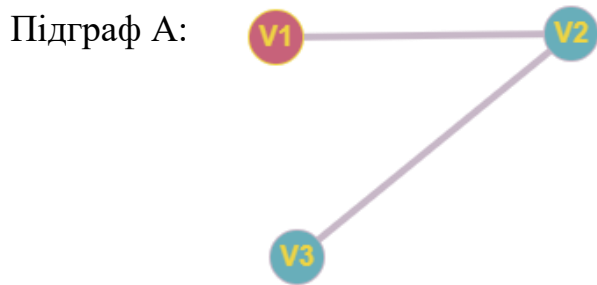
3) Кільцева сума $G1$ та $G2$ ($G1+G2$):



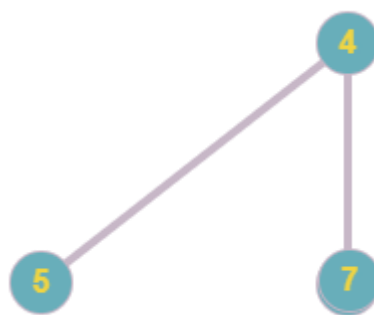
4) Розщеплення вершини $V1$ у другому графі:



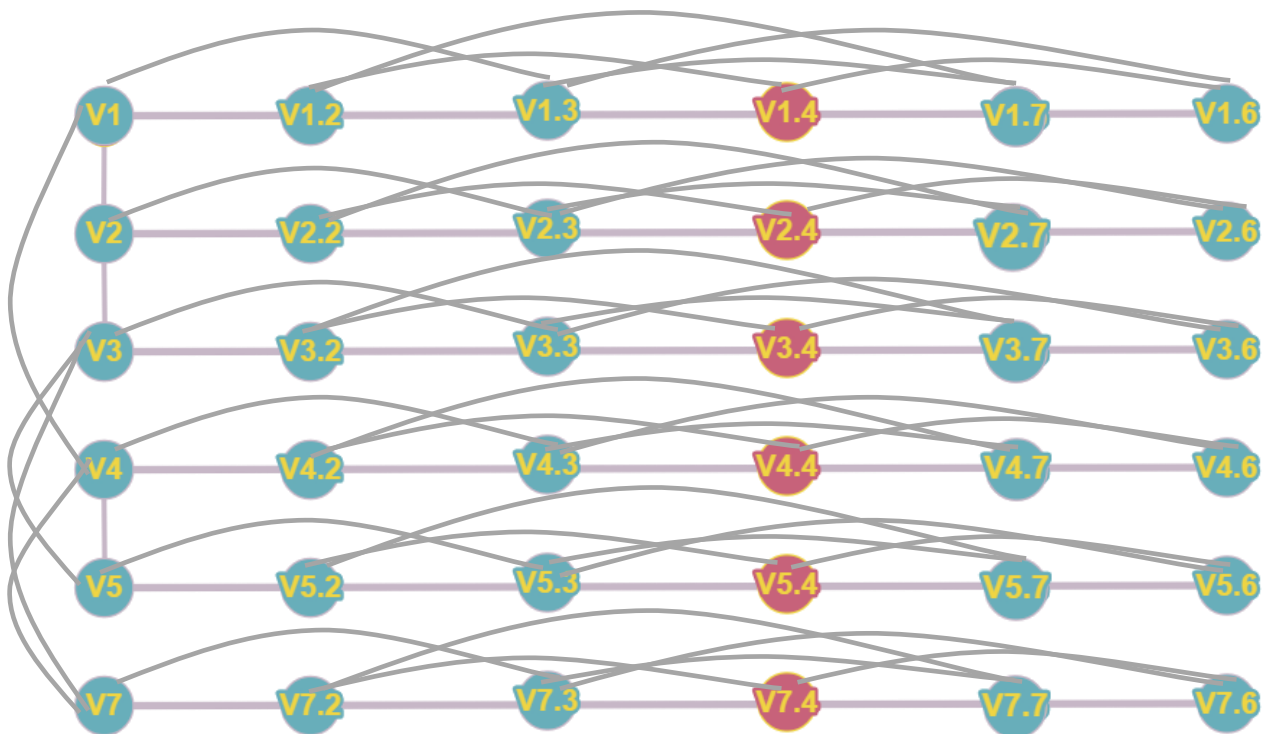
- 5) виділити підграф A, що складається з 3-х вершин в G1 і знайти стягнення A в G1 ($G1 \setminus A$):



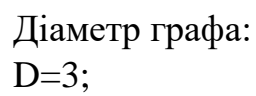
Стягнення A в G1:



- б) Добуток графів:



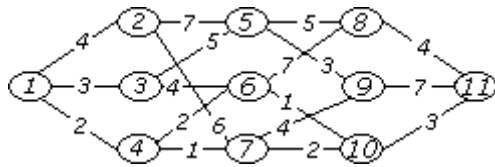
Знайти таблицю суміжності та діаметр графа.



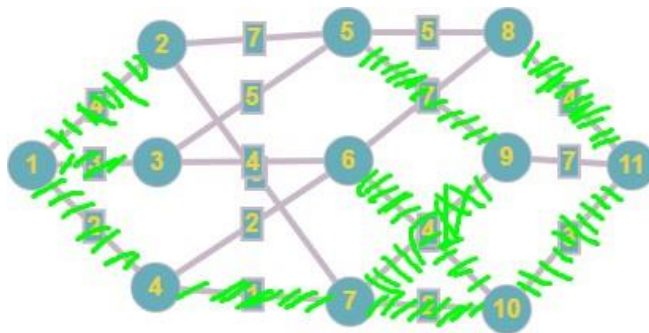
	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
V1	0	1	1	0	0	0	1	0	0	0
V2	1	0	1	0	1	0	1	0	0	0
V3	1	1	0	1	0	0	0	0	1	1
V4	0	0	1	0	1	0	0	1	0	1
V5	0	1	0	1	0	1	0	0	0	1
V6	0	0	0	0	1	0	1	1	0	1
V7	1	1	0	0	0	1	0	1	0	1
V8	0	0	0	1	0	1	1	0	0	0
V9	0	0	1	0	0	0	0	0	0	0
V10	0	0	1	1	1	1	1	0	0	0

Завдання № 3

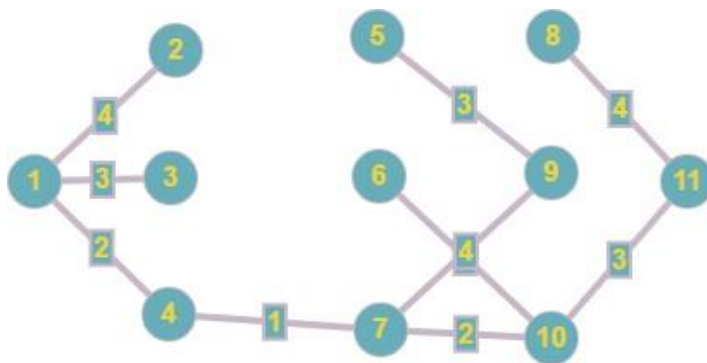
Знайти двома методами (Краскала і Прима) мінімальне остове дерево графа.



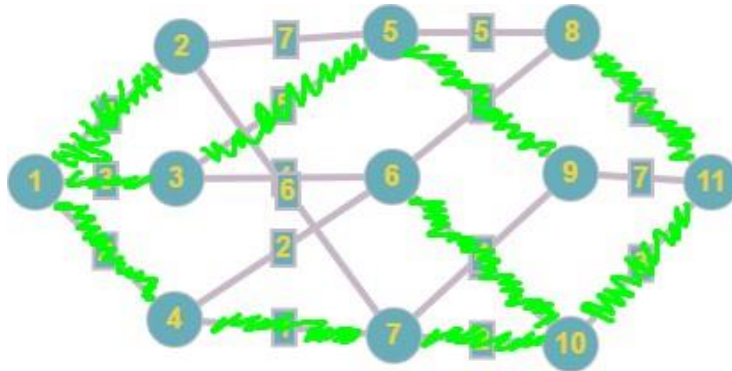
Метод Краскала:



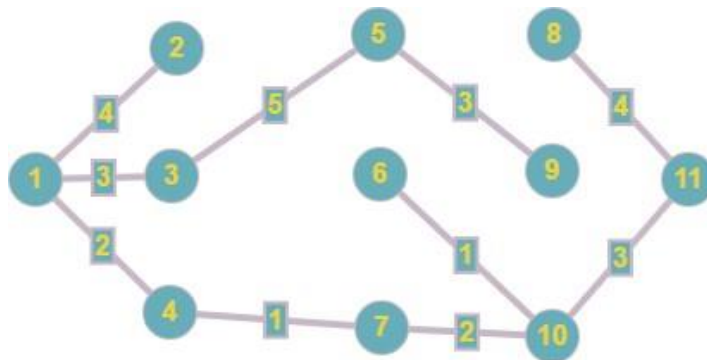
Шукане остове дерево:



Метод Прима:

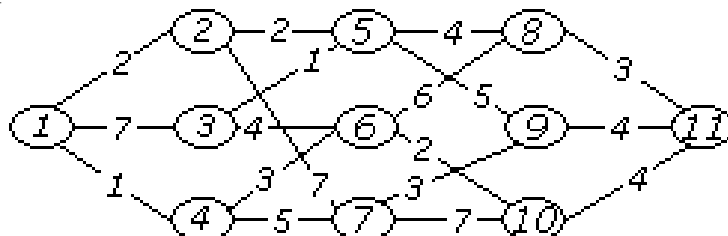


Шукане остове дерево:



Завдання 4:

За алгоритмом Прима знайти мінімальне остове дерево графа. Етапи розв'язання задачі виводити на екран. Протестувати розроблену програму на наступному графі:



Розв'язок:

```
#include<stdio.h>

typedef struct
{
```

```

    int first_vertex;

    int second_vertex;

    int weight;
}edge;

int whether_in_array(int arr[],int size,int element)
{
    for(int i=0; i<size; i++)
    {
        if(arr[i]==element)
        {
            return 1;
        }
    }
    return 0;
}

edge min_weight( edge a[], int lenght)
{
    edge min;
    min = a[0];
    for(int i=0; i<lenght; i++)
    {
        if(a[i].weight<min.weight)
        {
            min=a[i];

```



```

        }
    }

    return min;
}

int isinArray(int arr[],int size, int vertex )
{
    for(int i=0; i<size; i++)
    {
        if(arr[i] == vertex)
        {
            return 1;
        }
    }
    return 0;
}

int main(void)
{
    edge edges[18];
    for(int i=0;i<18;i++)
    {
        printf("Введіть першу вершину ребра: %d \n",i);
        scanf("%d", &edges[i].first_vertex);
        printf("Введіть другу вершину ребра: %d \n",i);
    }
}

```

```

scanf("%d", &edges[i].second_vertex);

printf("Введіть вагу ребра %d:\n",i);

scanf("%d", &edges[i].weight);

}

printf("\n");

for(int i=0;i<18;i++)

{

    printf("Ребро %d-%d має вагу\n",edges[i].first_vertex,edges[i].second_vertex,edges[i].weight);

}

printf("\n\n");

int vertexes[36];

for(int i=0; i<18; i++)

{

    vertexes[i*2]=edges[i].first_vertex;

    vertexes[i*2+1]=edges[i].second_vertex;

}

printf("\n");

int ult_vertexes[11];

int count=0;

for(int i=0; i<36; i++)

{

```

```

        if(!whether_in_array(ult_vertexes,count,vertexes[i]))
        {
            ult_vertexes[count]=vertexes[i];

            count++;
        }
    }

    printf("\n");

    int active_vertexes_arr[11];

    active_vertexes_arr[0]=ult_vertexes[0];

    int vcount=1;

    int active_ecount=0;

    printf("Алгоритм Прима:\n");

    do
    {
        edge edges_selection[11];

        int ecount=0;

        for(int i=0; i<18;i++)
        {
            if((isArray(active_vertexes_arr,vcount,edges[i].first_vertex) +
            isArray(active_vertexes_arr,vcount,edges[i].second_vertex)) % 2)

            {

                edges_selection[ecount]=edges[i];

                ecount++;
            }
        }
    }

```

```

    }

    edge min=min_weight(edges_selection,ecount);

    active_ecount++;

    if(isinArray(active_vertexes_arr,vcount,min.first_vertex))

    {

        active_vertexes_arr[vcount]=min.second_vertex;

    }

    else

    {

        active_vertexes_arr[vcount]=min.first_vertex;

    }

    printf("%d-%d |",min.first_vertex,min.second_vertex);

    vcount++;

}while(vcount!=11);

printf("\n");

}

```

Висновок:

На лабораторній роботі я набув практичних вмінь та навичок з використання алгоритмів Прима та Краскала.