

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

Кафедра систем штучного інтелекту

Звіт
Лабораторна робота №4
з дисципліни
«Дискретна математика»

Виконав:
студент групи КН-111
Жигайло Ярослав
Викладач:
Бойко Н.І.

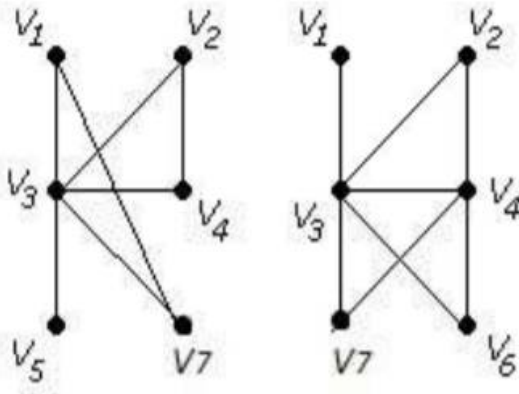
Львів – 2018 р.

ВАРІАНТ 7

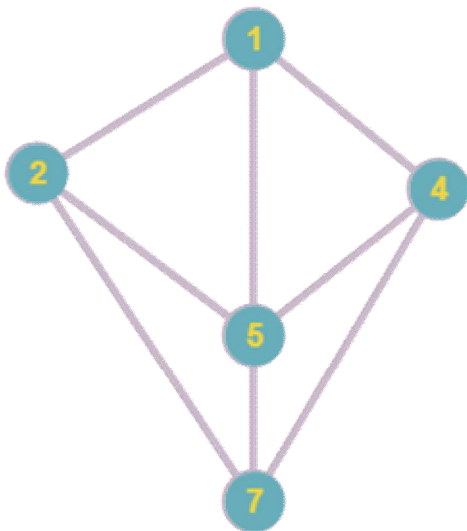
Завдання 1.

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.

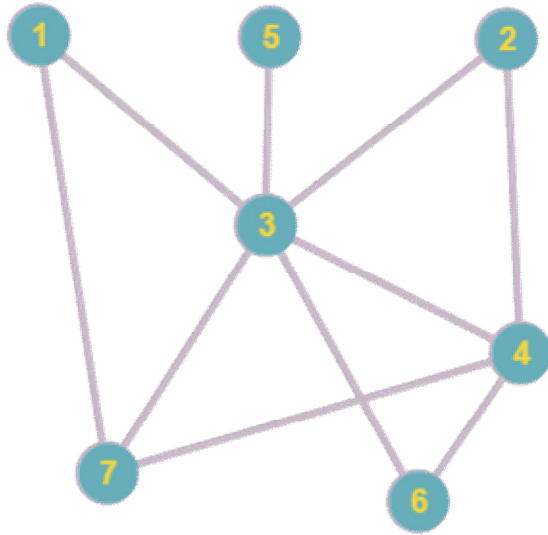
Початкові Графи :



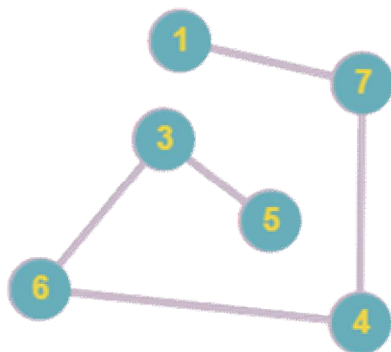
1)



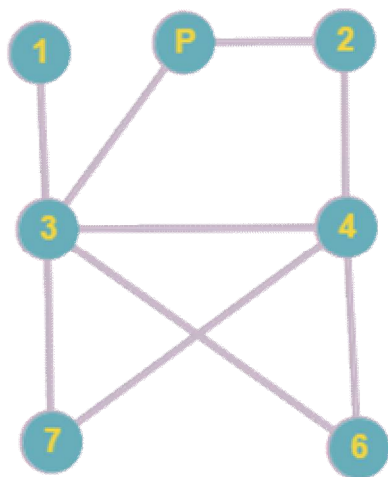
2)



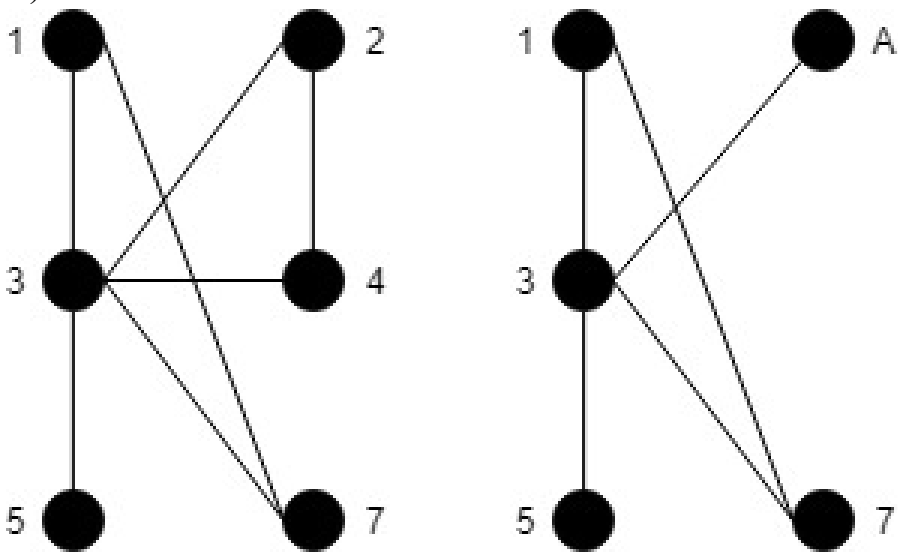
3)



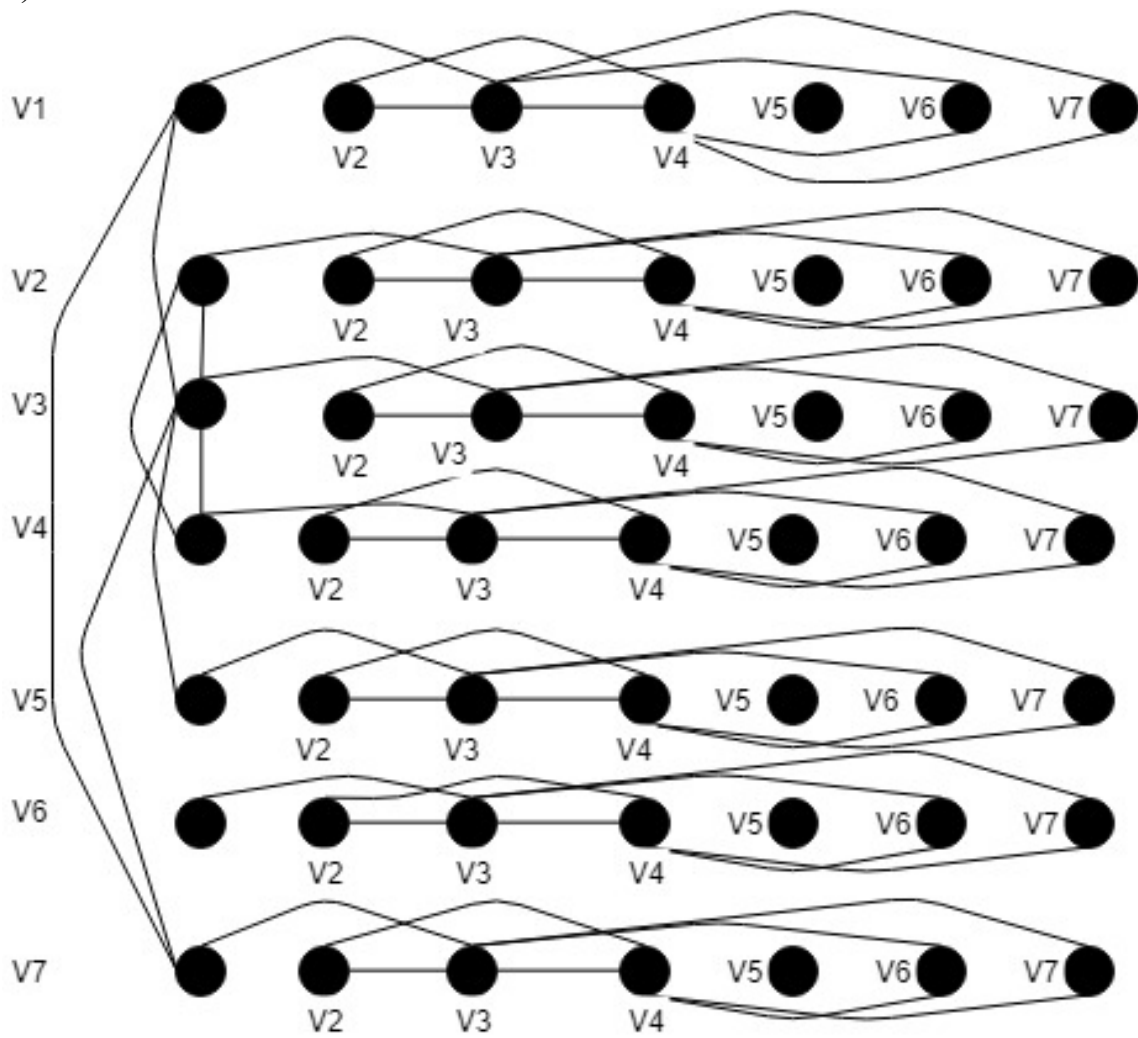
4)



5)

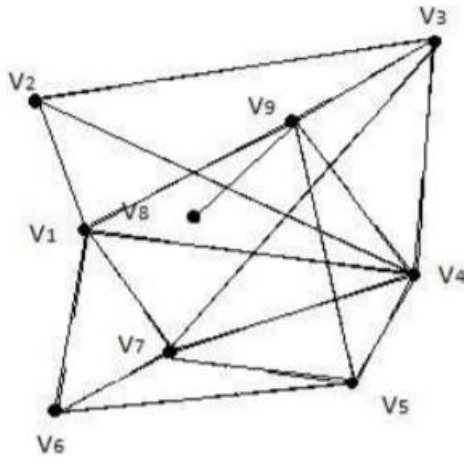


6)



Завдання 2

Знайти таблицю суміжності та діаметр графа.



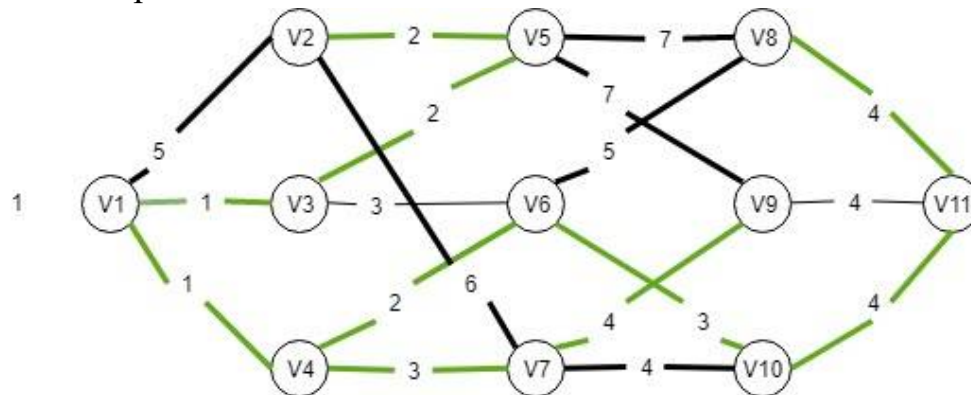
ТАБЛИЦЯ СУМІЖНОСТІ:

	V1	v2	v3	v4	v5	v6	v7	v8	v9
v1	0	1	0	1	0	1	0	0	0
v2	1	0	1	1	0	0	0	0	0
v3	0	1	0	1	0	0	1	0	1
v4	1	1	1	0	1	0	1	0	1
v5	0	0	0	1	0	1	1	0	1
v6	1	0	0	0	1	0	1	0	0
v7	0	0	1	1	1	1	0	0	0
v8	0	0	0	0	0	0	0	0	1
v9	0	0	1	1	1	0	0	1	0

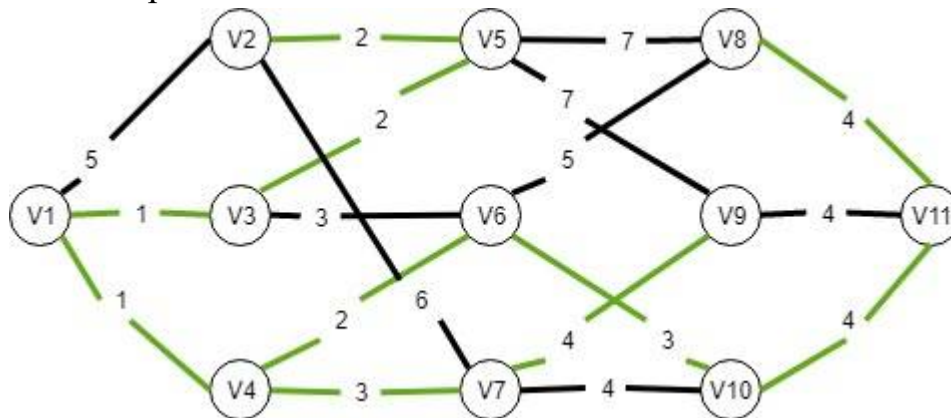
$d = 5$.

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

Метод Краскала:

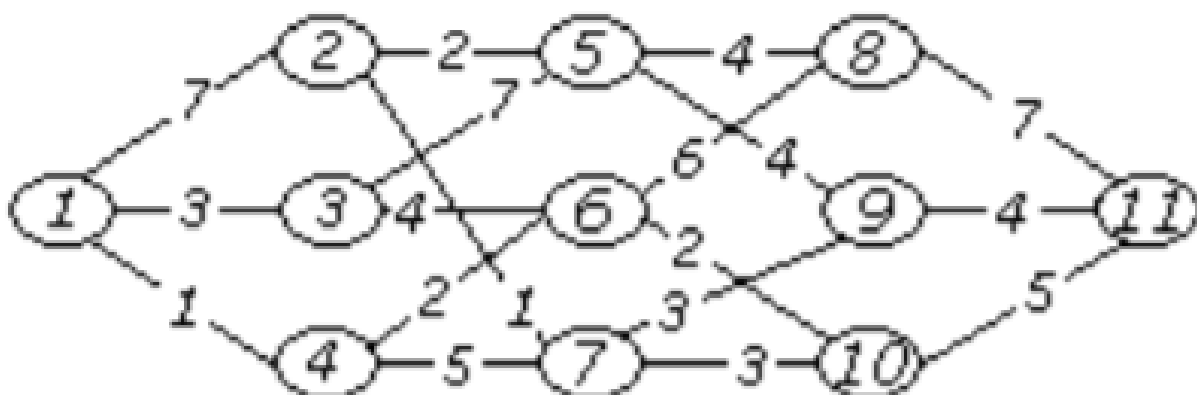


Метод Прима:



Як бачимо, в обох випадках мінімальне остове дерево графа однакове, бо воно є єдиним.

Завдання 4. За алгоритмом Прима знайти мінімальне остове дерево графа. Етапи розв'язання задачі виводити на екран. Протестувати розроблену програму на наступному графі.



Результат роботи програми:

```
The array of edges with weight:
Edge 1-2 has weight 7
Edge 1-3 has weight 3
Edge 1-4 has weight 1
Edge 2-5 has weight 2
Edge 5-8 has weight 4
Edge 8-11 has weight 7
Edge 10-11 has weight 5
Edge 7-10 has weight 3
Edge 4-5 has weight 7
Edge 3-6 has weight 4
Edge 3-5 has weight 7
Edge 2-7 has weight 1
Edge 4-6 has weight 2
Edge 5-9 has weight 4
Edge 6-8 has weight 6
Edge 6-10 has weight 2
Edge 7-9 has weight 3
Edge 9-11 has weight 4

The array of all the vertexes:
1|2|1|3|1|4|2|5|5|8|8|11|10|11|7|10|4|5|3|6|3|5|2|7|4|6|5|9|6|8|6|10|7|9|9|11|
The array of vertexes without repeats:
1|2|3|4|5|8|11|10|7|6|9|

The Pryma`s algorith graph:
1-4 |4-6 |6-10 |1-3 |7-10 |2-7 |2-5 |7-9 |5-8 |9-11 |
```

Код програми :

```
#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("Put the first vertex of %d edge:\n",i+1);
        scanf("%d", &edges[i].first_vertex);
        printf("Put the second vertex of %d edge:\n",i+1);
        scanf("%d", &edges[i].second_vertex);
        printf("Put the weight of %d edge:\n",i+1);
        scanf("%d", &edges[i].weight);

    }
    printf("The array of edges with weight:");
    printf("\n");
    for(int i=0;i<18;i++)
    {
        printf("Edge %d-%d has weight %d\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("The array of all the vertexes:\n");
for(int i=0; i<36; i++)
{
    printf("%d",vertexes[i]);
    printf("|");
}
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("The array of vertexes without repeats:\n");
for(int i=0; i<11; i++)
{
    printf("%d",ult_vertexes[i]);
    printf("|");
}
printf("\n");
printf("\n");
int active_vertexes_arr[11];
active_vertexes_arr[0]=ult_vertexes[0];
int vcount=1;
int active_ecount=0;
printf("The Pryma`s algorithim graph:\n");
do
{
    edge edges_selection[11];
    int ecount=0;
    for(int i=0; i<18;i++)
    {
        if((isinArray(active_vertexes_arr,vcount,edges[i].first_vertex)
isinArray(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");

}

```