

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

Кафедра систем штучного інтелекту

Звіт

Лабораторна робота №1

з дисципліни «Дискретна математика»

Виконав: студент групи КН-113

Власюк Олександр

Викладач: Мельникова Н.І.

Львів – 2019 р.

Тема Роботи:

Основні операції над графами. Знаходження остова мінімальної ваги за алгоритмом Прима-Краскала

Мета:

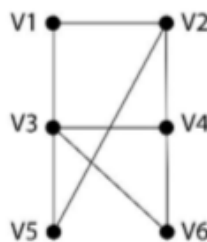
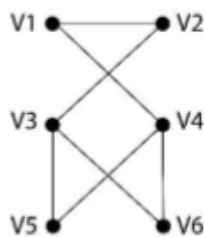
Набуття практичних вмінь та навичок з використання алгоритмів Прима і Краскала

Постановка завдання :

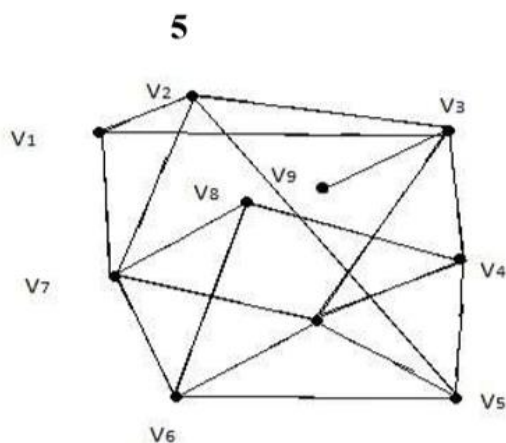
5-Варіант

1) Виконати наступні операції над графами:

- 1) знайти доповнення до першого графу,
- 2) об'єднання графів,
- 3) кільцеву суму $G1$ та $G2$,
- 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.

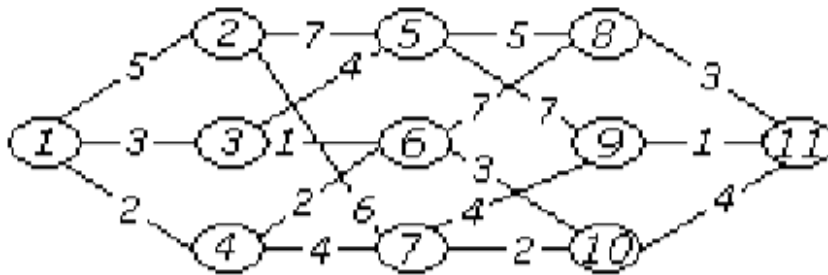


2. Знайти таблицю суміжності та діаметр графа.



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

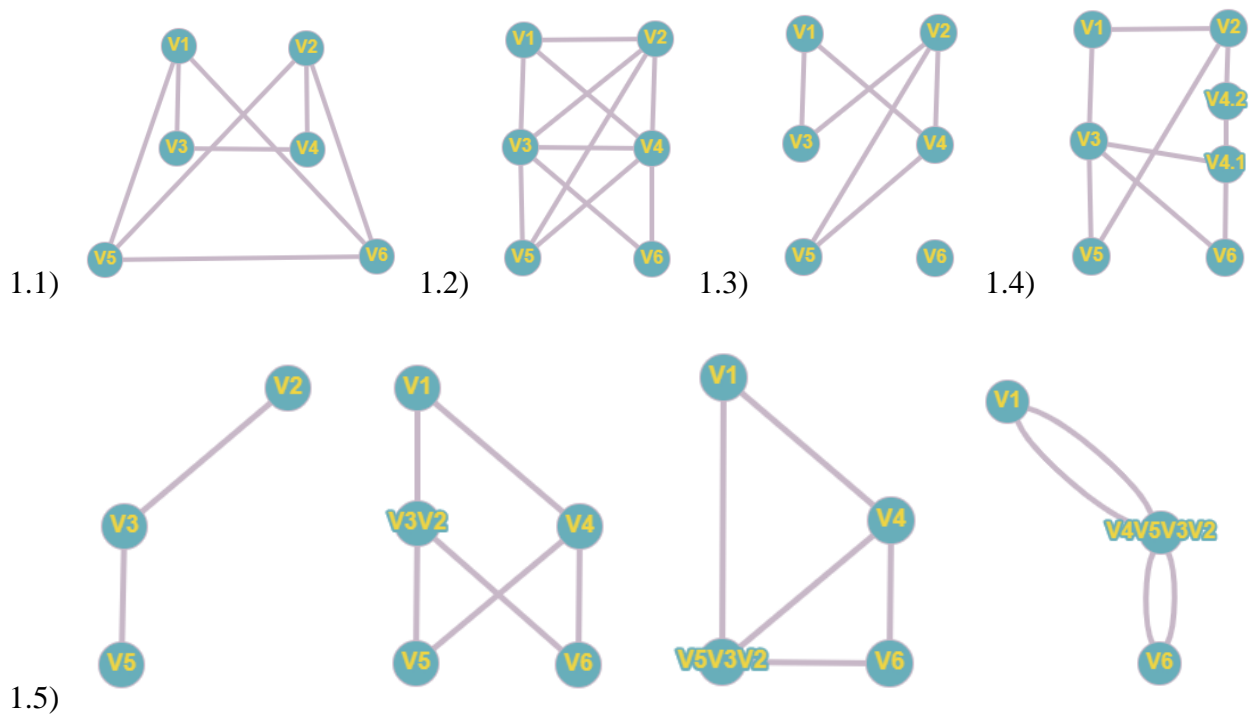
5



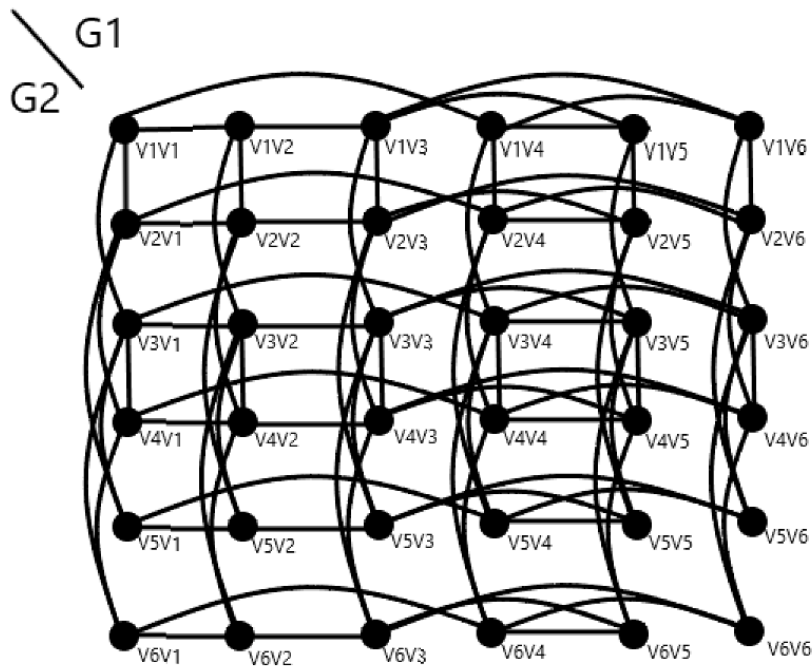
Розв'язки:

Завдання 1

Варіант 5



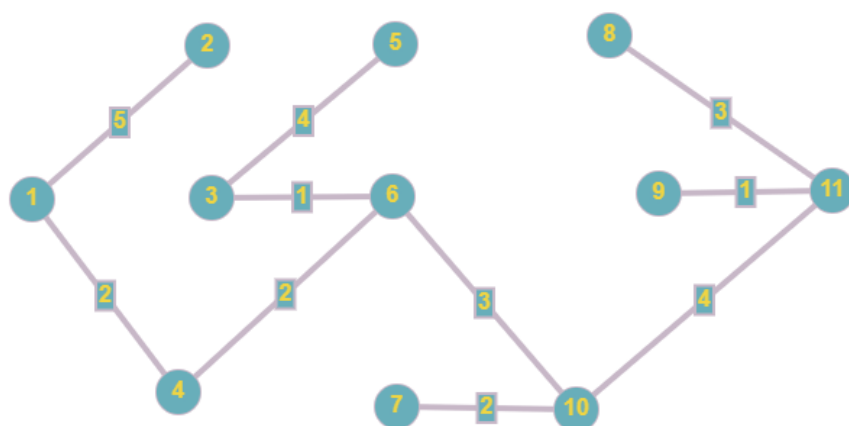
1.6) Добуток графів (наступна сторінка.)



2 ЗАВДАННЯ

Діаметр графа = 3

	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



3.

Алгоритм Краскала

$V = \{1, 4, 3, 5, 2, 6, 10, 7, 9, 8, 11, \}$

$E = \{(1,4), (3,5), (1,2), (2,5), (6,10), (4,6), (7,9), (8,11)\}$

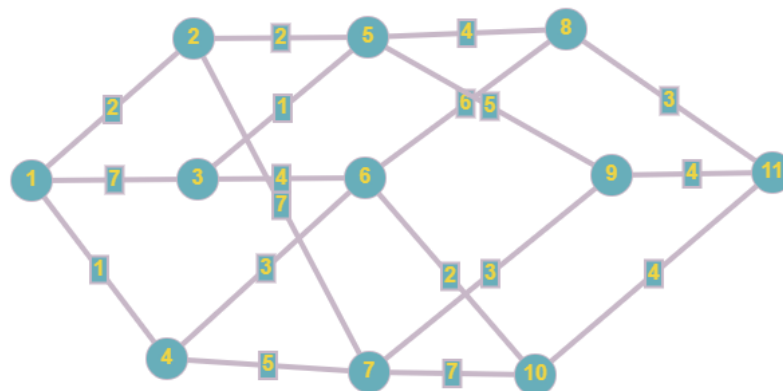
Алгоритм Прима

$V = \{1, 4, 2, 5, 3, 6, 10, 8, 11, 9, 7\}$

$E = \{(1,4), (1,2), (2,5), (3,5), (4,6), (6,10), (5,8), (8,11), (9,11), (7,9)\}$

Завдання 2 варіант №5

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



Код програмної реалізації:

```
#include <iostream>
```

```
using namespace std;
```

```
struct rebro
```

```
{
```

```
    int t1;
```

```
    int t2;
```

```
    int waga;
```

```
};
```

```
void vvid (rebro*p, int n, int m)
```

```
{
```

```
    cout<<"Vvedit' rebra(1 vershyna | 2vershyna | waga):\n";
```

```
    for (int i=0; i<m; i++)
```

```
    {
```

```
        cout<<"rebro"<<i+1<<": ";
```

```
        cin>>p[i].t1>>p[i].t2>>p[i].waga;
```

```
        while (p[i].t1<0 || p[i].t1>n || p[i].t2<0 || p[i].t2>n || p[i].waga<0)
```

```
        {
```

```
            cout<<"Nepravylni dani"<<endl;
```

```
            cout<<"vvedit dani zanovo"<<endl;
```

```
            cin>>p[i].t1>>p[i].t2>>p[i].waga;
```

```
        }
```

```
    }
```

```
}
```

```
void bulb(rebro*p, int n)
```

```
{
```

```
    rebro temp;
```

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

```
    {
```

```
        for (int j=0; j<n-i-1; j++)
```

```
        {
```

```
            if (p[j].waga > p[j+1].waga)
```

```

        {
            temp = p[j];
            p[j] = p[j+1];
            p[j+1] = temp;

        }
    }
}

```

```

bool vkluchene(int* a, int n, int f)

```

```

{
    for (int i=0; i<n; i++)
    {
        if (f == a[i])
        {
            return true;
        }
    }
    return false;
}

```

```

bool minn(int w, rebro* ed, int m, int* v, int n)

```

```

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

```

```

        if(((!vkluchene(v, n, ed[j].t1) && vkluchene(v, n, ed[j].t2)) ||
            (vkluchene(v, n, ed[j].t1) && !vkluchene(v, n, ed[j].t2)))
            && ed[j].waga < w)
        {
            return false;
        }
    }
    return true;
}

```

```

void pryma (rebro* ed, int*v, rebro*tree, int n, int m, int&i, int&j)
{
    if (i==n)
    {
        return;
    }
    else if (j==n)
    {
        j=1;
        vkluchene(v, n, ed[j].t2);
    }
    if(vkluchene(v, n, ed[j].t1) && vkluchene(v, n, ed[j].t2))
    {
        j++;
        pryma(ed, v, tree, n, m, i, j);
    }
}

```



```

    }
    else if (!(vkluchene(v, n, ed[j].t1)) && vkluchene(v, n, ed[j].t2)
        )
    {
        tree[i-1]=ed[j];
        v[i] = ed[j].t1;
        j++;
        i++;
        pryma(ed, v, tree, n, m, i, j);
    }

    else if (vkluchene(v, n, ed[j].t1) && !vkluchene(v, n, ed[j].t2) &&
minn(ed[j].waga, ed, m, v, n))
    {
        tree[i-1]=ed[j];
        v[i] = ed[j].t2;

        j++;
        i++;
        pryma(ed, v, tree, n, m, i, j);
    }
    else
    {
        j++;
        pryma(ed, v, tree, n, m, i, j);
    }
}

```

```

int main()
{
    int n,m;

    cout<<"skilky vershyn hochete vvesty?";

    cin>>n;

    cout<<"skilky reber hochete vvesty?";

    cin>>m;

    cout<<endl;

    rebro *ed = new rebro[m];

    int *v = new int[n];

    rebro *tree = new rebro [n-1];

    vvid (ed,n,m);

    bulb (ed,m);

    v[0]=ed[0].t1;

    v[1]=ed[0].t2;

    tree[0]=ed[0];

    int i=2;

    int j=1;

    pryma (ed,v,tree,n,m,i,j);

    cout<<"\nV = { ";

    for (int x=0; x<n; x++)

    {

        cout<<v[x]<<",";

    }

```

```

cout<<"}\nE = { ";

for (int x=0; x<n-1; x++)

{

    cout<<"{"<<tree[x].t2<<"} ";

}

cout<<"}\n";

return 0;

}

```

Результат виконаної програми:

```

kilky vershyn hochete vvesty?11
kilky reber hochete vvesty?18

/vedit' rebra(1 vershyna | 2vershyna | waga):
rebro1: 1 2 2
rebro2: 1 3 7
rebro3: 1 4 1
rebro4: 2 7 7
rebro5: 2 5 2
rebro6: 3 5 1
rebro7: 3 6 4
rebro8: 4 6 3
rebro9: 4 7 5
rebro10: 5 8 4
rebro11: 5 9 5
rebro12: 6 10 2
rebro13: 6 8 6
rebro14: 7 9 3
rebro15: 7 10 7
rebro16: 8 11 3
rebro17: 9 11 4
rebro18: 10 11 4

/ = { 1,4,2,5,3,6,10,8,11,9,7,}
: = { {1;4} {1;2} {2;5} {3;5} {4;6} {6;10} {5;8} {8;11} {9;11} {7;9} }

Process returned 0 (0x0)   execution time : 91.909 s
Press any key to continue.

```

Висновок:

Виконуючи лабораторну роботу я попрактикувався проводити різні операції з ними та у використанні алгоритмів Прима та Краскала.