

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Розрахункова робота

з дисципліни

«Дискретна математика»

Виконав:

Дребот Владислав Олегович

студент групи КН-112

Викладач:

Мельникова Н.І.

Львів – 2019 р.

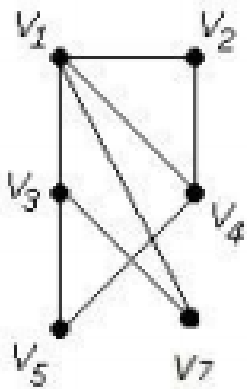
Варіант № 9

ІНДИВІДУАЛЬНІ ЗАВДАННЯ

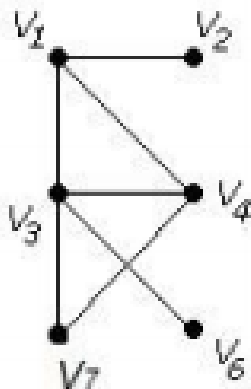
Завдання № 1

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.

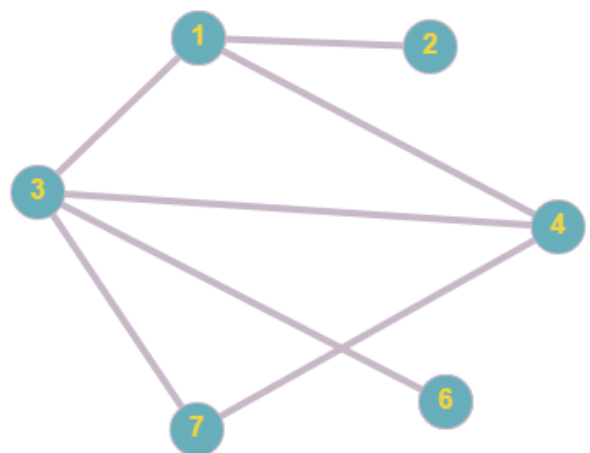
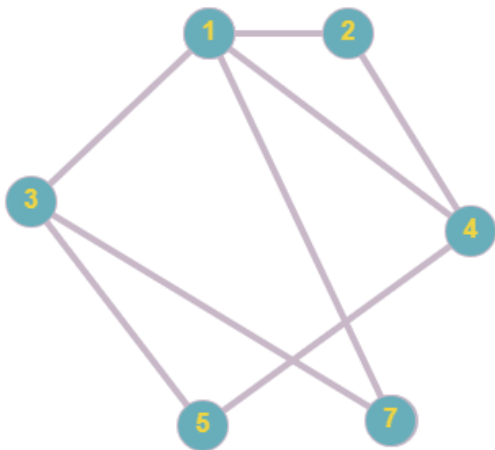
9)

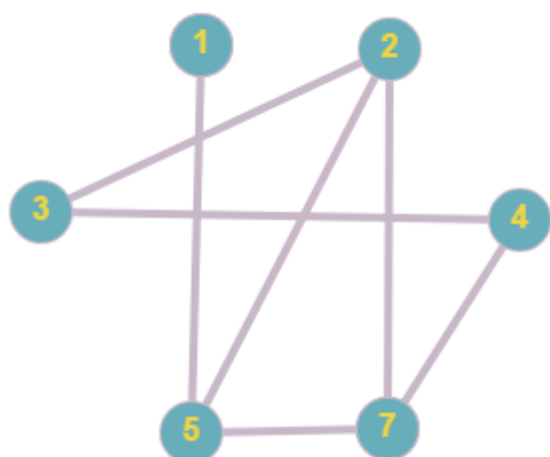


G1

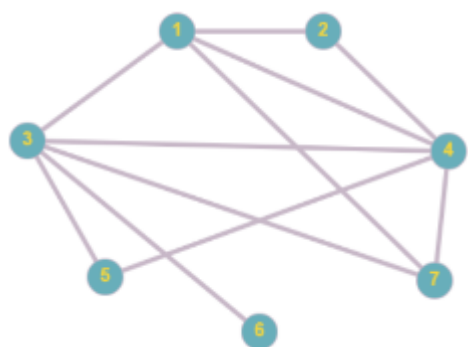


G2



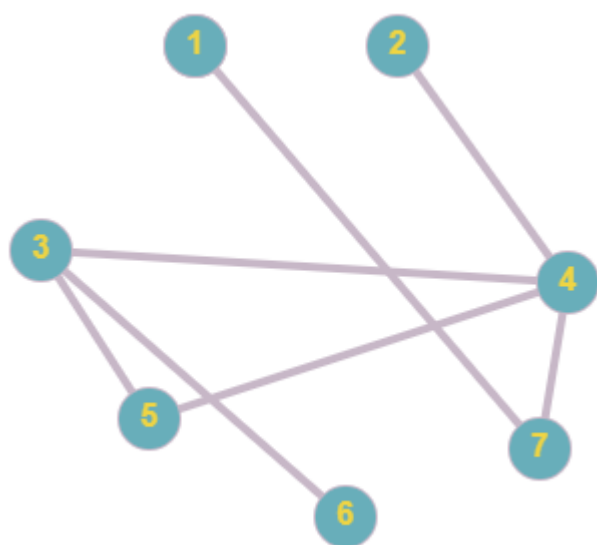


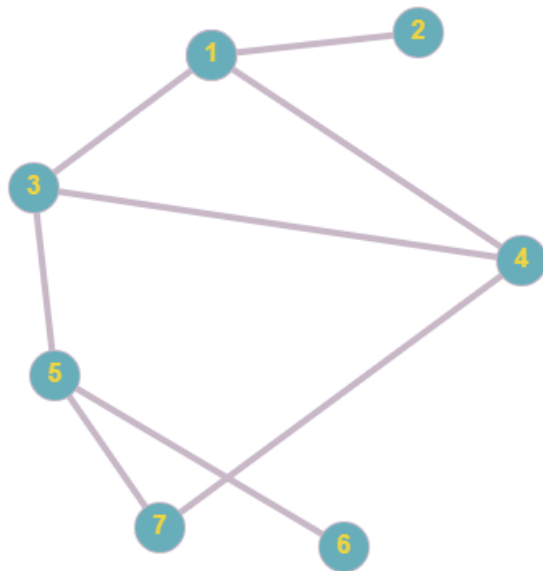
1)



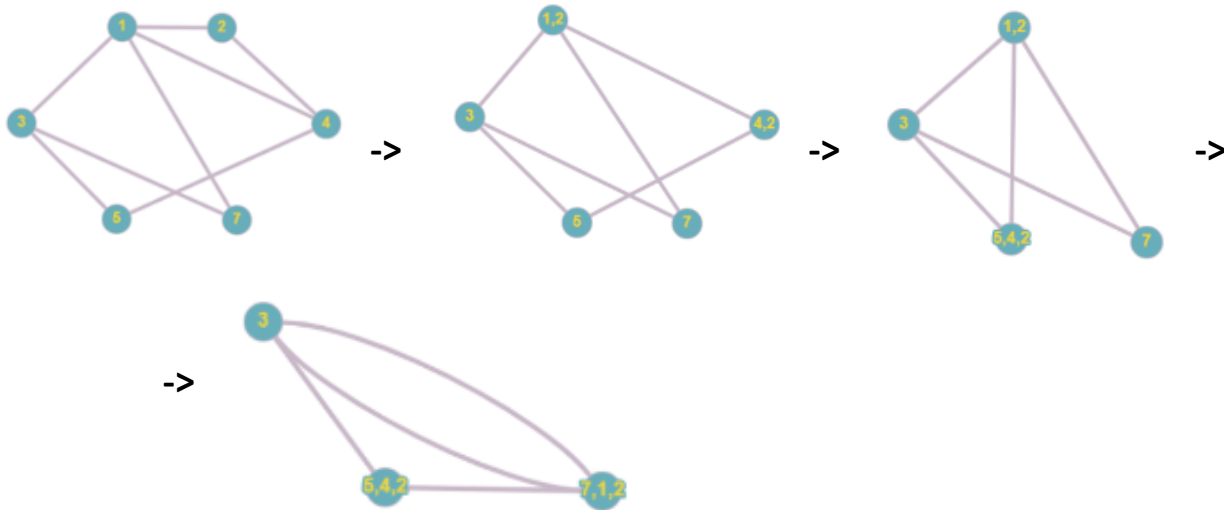
2)

3)





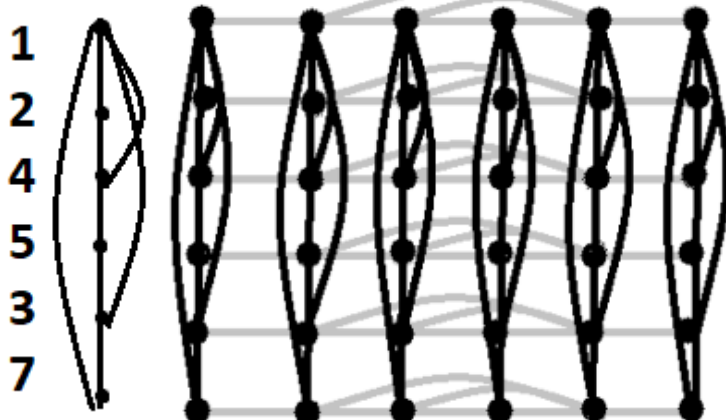
4)
5) G1



$G_a = (V, E)$

$V(G_a) = \{1, 2, 4\}$

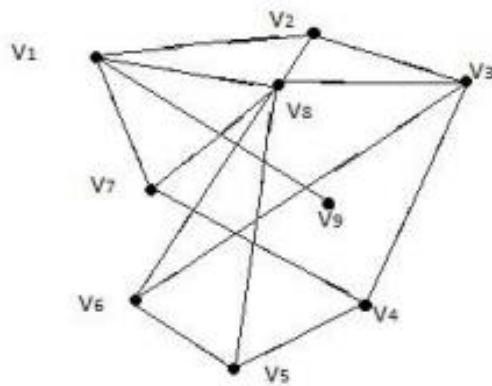
$V(G_1) = \{1, 2, 3, 4, 5, 7\}$



Завдання № 2

Скласти таблицю суміжності для орграфа.

9)



	1	2	3	4	5	6	7	8	9
1	0	1	0	0	0	0	1	1	1
2	1	0	1	0	0	0	0	1	0
3	0	1	0	1	0	1	0	1	0
4	0	0	1	0	1	0	1	0	0
5	0	0	0	1	0	1	0	1	0
6	0	0	1	0	1	0	0	1	0
7	1	0	0	1	0	0	0	1	0
8	1	1	1	0	1	1	1	0	0
9	1	0	0	0	0	0	0	0	0

Матриця ваг

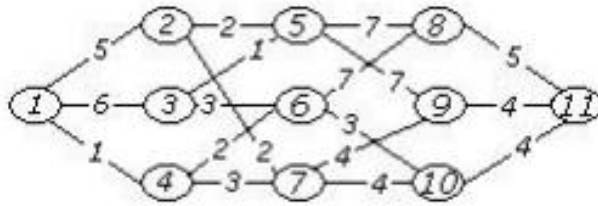
	1	2	3	4	5	6	7	8	9
1	0	1	2	2	2	2	1	1	1
2	1	0	1	2	2	2	2	1	2
3	2	1	0	1	2	1	2	1	3
4	2	2	1	0	1	2	1	2	3
5	2	2	2	1	0	1	2	1	3
6	2	2	1	2	1	0	2	1	3
7	1	2	2	1	2	2	0	1	2
8	1	1	1	2	1	1	1	0	2
9	1	2	3	3	3	3	2	2	0

Діаметр=3

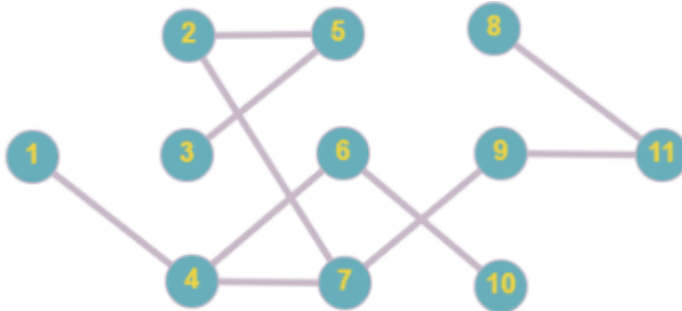
Завдання №5

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

9)



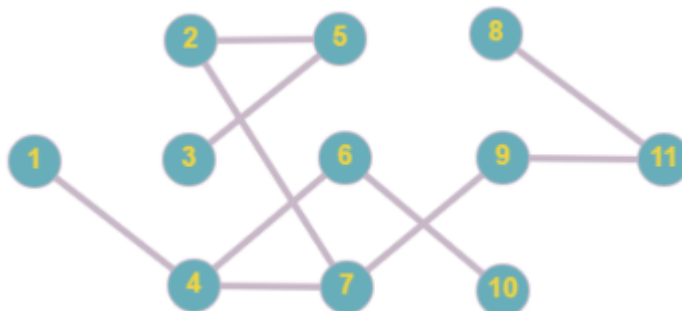
1)Прима



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

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

2)Краскала



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

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

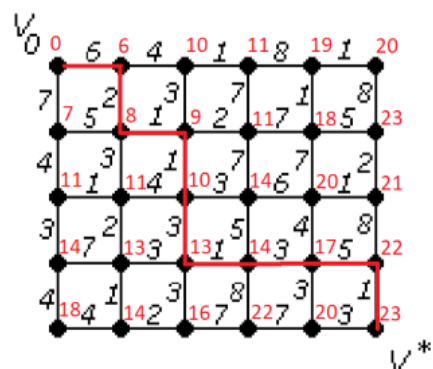
Завдання № 6

Розв'язати задачу комівояжера для повного 8-ми вершинного графа методом «іди у найближчий», матриця вагів якого має вигляд:

	1	2	3	4	5	6	7	8
1	∞	5	5	3	3	4	4	1
2	5	∞	4	3	2	1	4	6
3	5	4	∞	4	5	6	5	5
4	3	3	4	∞	1	5	1	7
5	3	2	5	1	∞	5	5	2
6	4	1	6	5	5	∞	7	3
7	4	4	5	1	5	7	∞	2
8	1	6	5	7	2	3	2	∞

Figure 1 shows a 5x5 grid of nodes. The top row is labeled v_0 . The values at the nodes are as follows:

	6	4	1	8	1
7	5	3	7	7	8
4	3	1	7	7	2
3	7	3	5	3	8
4	4	2	7	7	3



```

#include <iostream>
#define infinity 9999
using namespace std;

int minimum(int** arr, int v);
int main() {
    setlocale(LC_ALL, "Ukrainian");

    int v;
    cout << "Введіть кількість вершин графа : ";
    cin >> v;
    int** graph = new int* [v];
    for (int i = 0; i < v; i++) {
        graph[i] = new int[v];
    }
    for (int i = 0; i < v; i++) {
        for (int j = 0; j < v; j++) {
            graph[i][j] = 0;
        }
    }
    cout << "Введіть вагу ребер графа у формі 'V1 V2 вага', щоб завершити введіть 0 : "
    << endl;
    int a, b, c;
    while (true) {
        cin >> a;
        if (a == 0) {
            break;
        }
        cin >> b;
        cin >> c;
        graph[a - 1][b - 1] = graph[b - 1][a - 1] = c;
    }

    int** tops = new int* [v];
    for (int j = 0; j < v; j++) {
        tops[j] = new int[2];
    }
    int* path = new int[v];

    int start;
    cout << "Введіть вихідну вершину:";
    cin >> start;

    for (int i = 0; i < v; i++) {
        if (i == start - 1) {
            tops[i][0] = 0;
            tops[i][1] = 1;
        }
        else {
            tops[i][0] = infinity;
            tops[i][1] = 1;
        }
    }
    path[start - 1] = 0;

    int m;

    for (int i = 0; i < v; i++) {
        m = minimum(tops, v);
        for (int j = 0; j < v; j++) {
            if (graph[m][j]) {
                if (tops[j][0] > tops[m][0] + (graph[m][j])) {
                    tops[j][0] = tops[m][0] + (graph[m][j]);
                    path[j] = m;
                }
            }
        }
    }
}

```



```

        }
    }
    }
    tops[m][1] = 0;
}

cout << "Введіть потрібну вершину: ";
int final;
cin >> final;
cout << "Мінімальний шлях: ";
cout << tops[final-1][0];
cout << endl << final << " <- ";
final--;
while (path[final] != start - 1) {
    cout << path[final] + 1 << " <- ";
    final = path[final];
}
cout << start << endl;
return 0;
}

int minimum(int** arr, int v) {
    int m = 0;
    for (int i = 0; i < v; i++) {
        if (arr[i][1]) {
            m = i;
            break;
        }
    }
    for (int i = 1; i < v; i++) {
        if (arr[m][0] >= arr[i][0] && arr[i][1] == 1) {
            m = i;
        }
    }
    return m;
}
}

```

Результат:

```
Введіть кількість вершин графа : 30
Введіть вагу ребер графа у формі 'V1 V2 вага', щоб завершити введіть 0 :
1 2 6
2 3 1
3 4 3
4 5 1
5 6 3
7 8 4
8 9 1
9 10 1
10 11 2
11 12 7
13 14 4
14 15 1
15 16 2
16 17 3
17 18 7
19 20 7
20 21 3
21 22 8
22 23 1
23 24 5
25 26 4
26 27 7
27 28 3
28 29 3
29 30 6
```

```

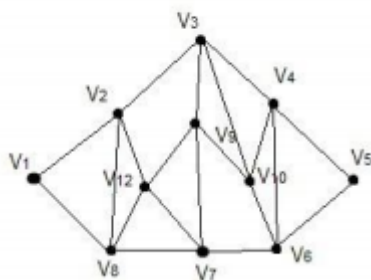
1 7 4
2 8 2
3 9 3
4 10 4
5 11 5
6 12 8
7 13 4
8 14 1
9 15 5
10 16 7
11 17 1
12 18 3
13 19 5
14 20 1
15 21 7
16 22 7
17 23 2
18 24 8
19 25 7
20 26 2
21 27 3
22 28 1
23 29 3
24 30 8
0
Введіть вихідну вершину: 1
Введіть потрібну вершину: 30
Мінімальний шлях: 24
30 <- 29 <- 23 <- 17 <- 11 <- 10 <- 9 <- 8 <- 7 <- 1

```

Завдання № 8

Знайти ейлеровий цикл в ейлеровому графі двома методами: а) Флері; б) елементарних циклів.

9)



a) 1-2-8-11-9-10-6-4-10-3-9-7-11-2-3-4-5-6-7-8-1

```

#include<iostream>
#include<vector>
using namespace std;
void Search(int v, vector < vector<int> >* G, int N)
{
    int i;
    for (i = 0; i < N; i++)
        if ((*G)[v][i])
        {
            (*G)[v][i] = (*G)[i][v] = 0;
            Search(i, G, N);
        }
    cout << v + 1 << " ";
}

int main()
{
    int N = 0;
    cout << "Enter N:"; cin >> N;
    vector < vector<int> > G(N, vector<int>(N));
    for (int i = 0; i < N; ++i)
        for (int j = 0; j < N; ++j)
            cin >> G[i][j];
    int Node_power, p, q, sum;
    Node_power = 1;
    for (p = 0; p < N; p++)
    {
        sum = 0;
        for (q = 0; q < N; q++)
        {
            sum += G[p][q];
        }
        if (sum % 2) Node_power = 0;
    }
}

```

```

Node_power = 1;
for (p = 0; p < N; p++)
{
    sum = 0;
    for (q = 0; q < N; q++)
    {
        sum += G[p][q];
    }
    if (sum % 2) Node_power = 0;
}
cout << endl;
if (Node_power)
    Search(0, &G, N);
else
    cout << "not Eulerian graph\n";
cout << endl;
return 0;
}

```

Консоль отладки Microsoft Visual Studio

```

Enter N:11
0 1 0 0 0 0 0 1 0 0 0
1 0 1 0 0 0 0 1 0 0 1
0 1 0 1 0 0 0 0 1 1 0
0 0 1 0 1 1 0 0 0 1 0
0 0 0 1 0 1 0 0 0 0 0
0 0 0 1 1 0 1 0 0 1 0
0 0 0 0 0 1 0 1 1 0 1
1 1 0 0 0 0 1 0 0 0 1
0 0 1 0 0 0 1 0 0 1 1
0 0 1 1 0 1 0 0 1 0 0
0 1 0 0 0 0 1 1 1 0 0

```

```

1 8 11 7 8 2 11 9 10 6 7 9 3 10 4 6 5 4 3 2 1

```

Результат:

6)

1) 1 – 2 – 8 – 1 +

2) 2 – 3 – 9 – 12 -2 +

3) 12 – 8 – 7 -12

4) 7 – 9 – 10 – 6 – 7

5) 3 – 10 - 4 - 3

5) 6 – 4 – 5 – 6

• 1 – 2 – 8 – 1

• 1 – 2 – 3 – 9 – 12 - 2 – 8 – 1

• 1 – 2 – 3 – 9 – 12 - 2 – 8- 12 – 7 – 9 – 10 – 6 – 7 – 8 – 1

• 1 – 2 – 3 – 9 – 12 - 2 – 8- 12 – 7 – 9 – 10 – 6 – 4 – 10 – 3 – 4
– 5 – 6 – 7 – 8 – 1

Завдання № 9

$$x \Rightarrow y = y \vee \neg x$$

$$y \Rightarrow z = z \vee \neg y$$

$$(x \Rightarrow y) \wedge (y \Rightarrow z) = (y \wedge z) \vee (\neg x \wedge \neg y)$$

$$x \Rightarrow z = z \vee \neg x$$

$$((x \Rightarrow y) \wedge (y \Rightarrow z)) \Rightarrow (x \Rightarrow z) = 1$$