

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

Кафедра систем штучного інтелекту

Лабораторна робота №5

з дисципліни
«Дискретна математика»

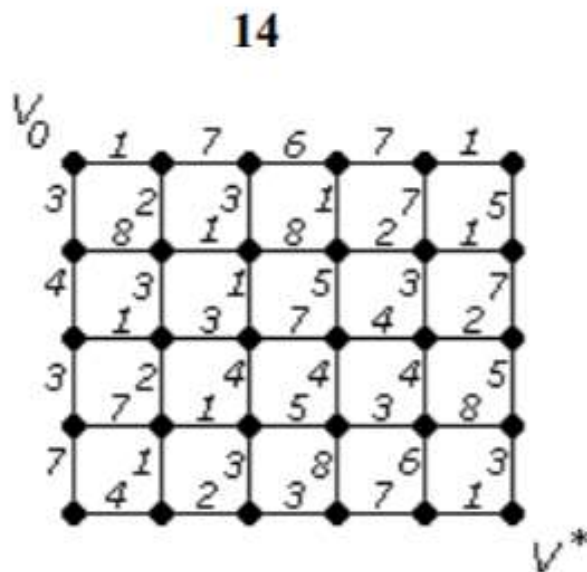
Виконав:
студент групи КН-114
Чорний Святослав
Викладач:
Мельникова Н.І.

Львів – 2019р.

Тема: Знаходження найкоротшого маршруту за алгоритмом Дейкстри.
Плоскі планарні графи
Мета роботи: набуття практичних вмінь та навичок з використання алгоритму Дейкстри.

Варіант 14 Індивідуальне завдання

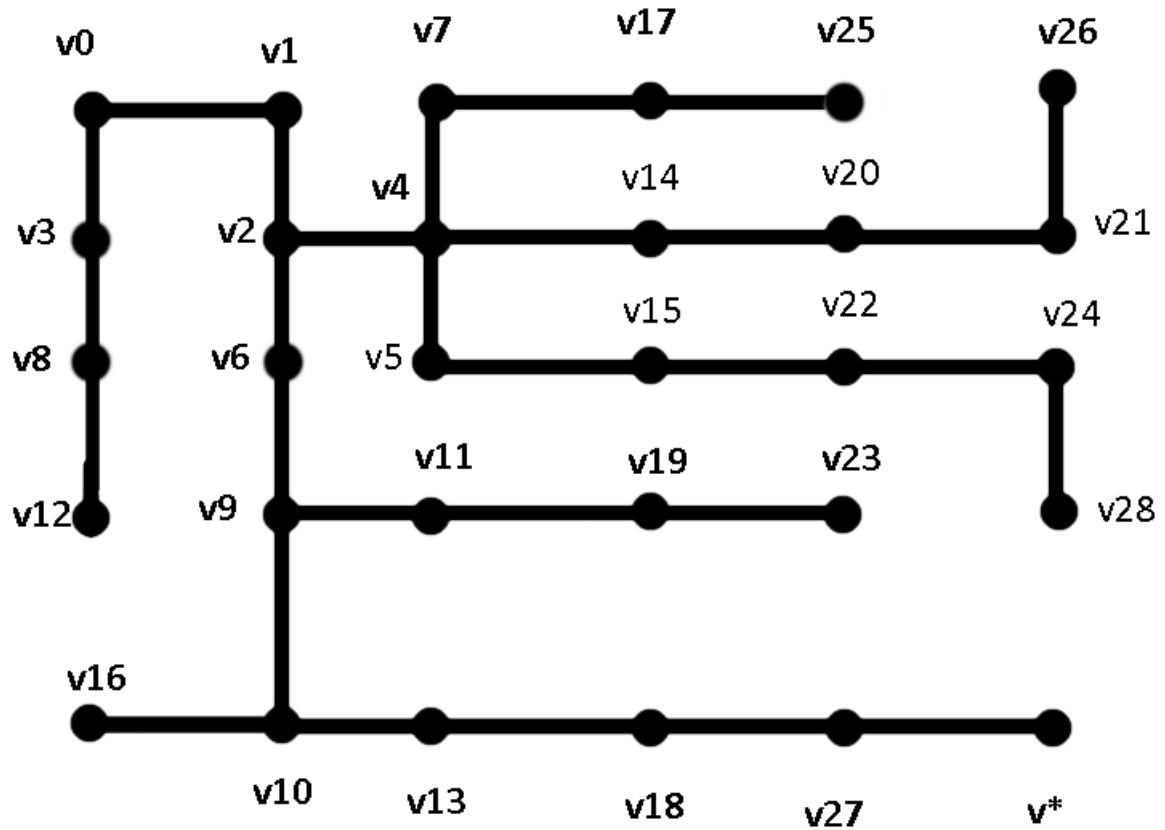
Завдання №1: За допомогою алгоритму Дейкстра знайти найкоротший шлях у графі поміж парою вершин V_0 і V^*



Кроки вирішення

- з вершини V_0 в в вершину $V_1 = 1$ $V_0 \rightarrow V_1$;
- з вершини V_0 в вершину $V_2 = 3$ $V_0 \rightarrow V_2$;
- з вершини V_0 в вершину $V_3 = 3$ $V_0 \rightarrow V_1 \rightarrow V_3$;
- з вершини V_0 в вершину $V_4 = 4$
 $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_4$;
- з вершини V_0 в вершину $V_5 = 5$
 $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_4 \rightarrow V_5$;
- з вершини V_0 в вершину $V_6 = 6$
 $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_6$;
- з вершини V_0 в вершину $V_7 = 7$
 $V_0 \rightarrow V_2 \rightarrow V_7$;
- з вершини V_0 в вершину $V_8 = 7$
 $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_4 \rightarrow V_8$;

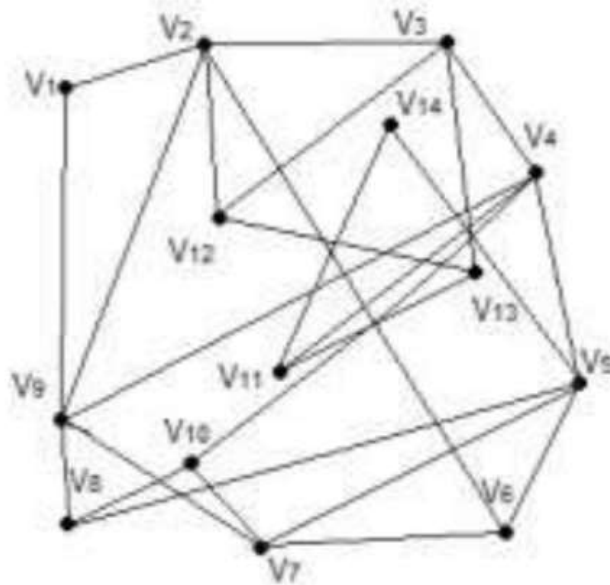
з вершини V_0 в вершину $V_9 = 8$
 $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_6 \rightarrow V_9$;
 з вершини V_0 в вершину $V_{10} = 9$
 $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_6 \rightarrow V_9 \rightarrow V_{10}$
 з вершини V_0 в вершину $V_{11} = 9$
 $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_6 \rightarrow V_9 \rightarrow V_{11}$;
 з вершини V_0 в вершину $V_{12} = 10$
 $V_0 \rightarrow V_2 \rightarrow V_7 \rightarrow V_{12}$;
 з вершини V_0 в вершину $V_{13} = 11$
 $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_6 \rightarrow V_9 \rightarrow V_{11} \rightarrow V_{13}$;
 з вершини V_0 в вершину $V_{14} = 12$
 $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_4 \rightarrow V_{14}$;
 з вершини V_0 в вершину $V_{15} = 12$
 $V_0 \rightarrow V_1 \rightarrow V_2 \rightarrow V_4 \rightarrow V_5 \rightarrow V_{15}$;
 з вершини V_0 в вершину $V_{16} = 13$
 $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_6 \rightarrow V_9 \rightarrow V_{11} \rightarrow V_{16}$;
 з вершини V_0 в вершину $V_{17} = 13$
 $V_0 \rightarrow V_1 \rightarrow V_7 \rightarrow V_8 \rightarrow V_9 \rightarrow V_{10} \rightarrow V_{16} \rightarrow V_{17}$;
 з вершини V_0 в вершину $V_{18} = 14$
 $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_6 \rightarrow V_9 \rightarrow V_{11} \rightarrow V_{13} \rightarrow V_{18}$;
 з вершини V_0 в вершину $V_{19} = 14$
 $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_6 \rightarrow V_9 \rightarrow V_{10} \rightarrow V_{19}$;
 з вершини V_0 в вершину $V_{20} = 14$
 $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_4 \rightarrow V_{14} \rightarrow V_{20}$;
 з вершини V_0 в вершину $V_{21} = 15$
 $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_4 \rightarrow V_{14} \rightarrow V_{20} \rightarrow V_{21}$;
 з вершини V_0 в вершину $V_{22} = 16$
 $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_4 \rightarrow V_5 \rightarrow V_{15} \rightarrow V_{22}$;
 з вершини V_0 в вершину $V_{23} = 17$
 $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_6 \rightarrow V_9 \rightarrow V_{10} \rightarrow V_{19} \rightarrow V_{23}$;
 з вершини V_0 в вершину $V_{24} = 18$
 $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_4 \rightarrow V_5 \rightarrow V_{15} \rightarrow V_{22} \rightarrow V_{24}$;
 з вершини V_0 в вершину $V_{25} = 20$
 $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_4 \rightarrow V_8 \rightarrow V_{17} \rightarrow V_{25}$;
 з вершини V_0 в вершину $V_{26} = 20$
 $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_4 \rightarrow V_{14} \rightarrow V_{20} \rightarrow V_{21} \rightarrow V_{26}$;
 з вершини V_0 в вершину $V_{27} = 21$

$$V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_6 \rightarrow V_9 \rightarrow V_{11} \rightarrow V_{13} \rightarrow V_{18} \rightarrow V_{27} \rightarrow V^*$$


Найменший шлях від вершини V_0 до $V^* = 22$.

Завдання №2: За допомогою γ -алгоритма зробити укладку графа у площині, або довести що вона неможлива.

14



Щоб укласти граф на площині він повинен бути планарним. Умова планарності- $V-E+F=2$

V-кількість вершин

E-кількість ребер

F-кількість граней

З цієї формули випливає інакше

$$E \leq 3V - 6$$

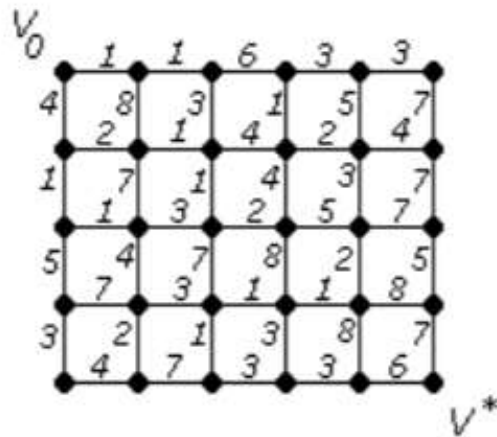
Підставляємо значення

$$26 \leq 36$$

Отже, цей граф не планарний

Завдання №2

Написати програму, яка реалізує алгоритм Дейкстри знаходження найкоротшого шляху між парою вершин у графі. Протестувати розроблену програму на графі згідно свого варіанту.



```

#define _CRT_SECURE_NO_WARNINGS
#include<iostream>

using namespace std;

int main()
{
    int maximalNumber = 10000;

    int vertex, edges; //к-сть вершин, ребер
    cout << " vertexes: ";
    cin >> vertex;
    cout << " edges: ";
    cin >> edges;
    int** a = new int* [vertex];
    int* d = new int[vertex];
    int* visited = new int[vertex];
    int temp, minindex, min;

    for (int i = 0; i < vertex; i++)
    {
        a[i] = new int[vertex];
    }

    for (int i = 0; i < vertex; i++)
    {
        for (int j = 0; j < vertex; j++) {
            a[i][j] = 0;
        }
    }

    a[0][1] = 1;
    a[0][6] = 4;
    a[1][0] = 1;
    a[1][2] = 1;
    a[1][7] = 8;
    a[2][1] = 1;

```

```
a[2][8] = 3;
a[2][3] = 6;
a[3][2] = 6;
a[3][4] = 3;
a[3][9] = 1;
a[4][3] = 3;
a[4][5] = 3;
a[4][10] = 5;
a[5][4] = 3;
a[5][11] = 7;
a[6][0] = 4;
a[6][7] = 2;
a[6][12] = 1;
a[7][1] = 8;
a[7][6] = 2;
a[7][8] = 1;
a[7][13] = 7;
a[8][7] = 1;
a[8][2] = 3;
a[8][14] = 1;
a[8][9] = 4;
a[9][8] = 4;
a[9][3] = 1;
a[9][15] = 4;
a[9][10] = 2;
a[10][9] = 2;
a[10][4] = 5;
a[10][16] = 3;
a[10][11] = 4;
a[11][10] = 4;
a[11][5] = 7;
a[11][17] = 7;
a[12][6] = 1;
a[12][13] = 1;
a[12][18] = 5;
a[13][12] = 1;
a[13][7] = 7;
a[13][19] = 4;
a[13][14] = 3;
a[14][13] = 3;
a[14][8] = 1;
a[14][20] = 7;
a[14][15] = 2;
a[15][14] = 2;
a[15][9] = 4;
a[15][21] = 8;
a[15][16] = 5;
a[16][15] = 5;
a[16][10] = 3;
a[16][22] = 2;
a[16][17] = 7;
a[17][16] = 7;
a[17][11] = 7;
a[17][23] = 5;
a[18][12] = 5;
a[18][19] = 7;
a[18][24] = 3;
a[19][18] = 7;
a[19][13] = 4;
a[19][25] = 2;
a[19][20] = 3;
a[20][19] = 3;
a[20][14] = 7;
a[20][26] = 1;
a[20][21] = 1;
```

```

a[21][20] = 1;
a[21][15] = 8;
a[21][27] = 8;
a[21][22] = 1;
a[22][21] = 1;
a[22][16] = 2;
a[22][28] = 8;
a[22][23] = 8;
a[23][22] = 8;
a[23][17] = 5;
a[23][29] = 7;
a[24][18] = 3;
a[24][25] = 4;
a[25][24] = 4;
a[25][19] = 2;
a[25][26] = 7;
a[26][25] = 7;
a[26][20] = 1;
a[26][27] = 3;
a[27][26] = 3;
a[27][21] = 8;
a[27][28] = 3;
a[28][27] = 3;
a[28][22] = 8;
a[28][29] = 6;
a[29][28] = 6;
a[29][23] = 7;

```

```

for (int i = 0; i < vertex; i++)
{
    for (int j = 0; j < vertex; j++)
        cout << a[i][j] << " ";
    cout << endl;
}

for (int i = 0; i < vertex; i++)
{
    d[i] = maximalNumber;
    visited[i] = 1;
}

int start, finish;
cout << "From vertex :";
cin >> start;
start--;
cout << "To: ";
cin >> finish;
finish--;
int begin_index = start;
d[begin_index] = 0;

do {
    minindex = maximalNumber;
    min = maximalNumber;
    for (int i = 0; i < vertex; i++)
    {
        if ((visited[i] == 1) && (d[i] < min))
        {
            min = d[i];
            minindex = i;
        }
    }
}

```



```

    }
    if (minindex != maximalNumber)
    {
        for (int i = 0; i < vertex; i++)
        {
            if (a[minindex][i] > 0)
            {
                temp = min + a[minindex][i];
                if (temp < d[i])
                {
                    d[i] = temp;
                }
            }
        }
        visited[minindex] = 0;
    }
} while (minindex < maximalNumber);

cout << "Minimal ways to vertex: " << endl;
for (int i = 0; i < vertex; i++)cout << d[i] << " ";

bool flag = false;
for (int i = 0; i < vertex; i++)if (d[i] != 0 && d[i] != maximalNumber)flag =
true;

if (flag) {
    int* ver = new int[vertex];
    int end = finish;
    ver[0] = end + 1;
    int k = 1;
    int weight = d[end];

    while (end != begin_index)
    {
        for (int i = 0; i < vertex; i++)
            if (a[end][i] != 0)
            {
                int temp = weight - a[end][i];
                if (temp == d[i])
                {
                    weight = temp;
                    end = i;
                    ver[k] = i + 1;
                    k++;
                }
            }
    }

    cout << endl << "Print minimal way" << endl;
    for (int i = k - 1; i >= 0; i--)cout << ver[i] << " ";
    cout << endl << "Min weight = " << d[finish-1];

}
else {
    cout << "There isnt such way";
}
return 0;
}

```

