

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА
ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота № 5

з дисципліни

«Дискретна математика»

Виконала:

студентка групи КН-112

Тимчишин Марта

Викладач:

Мельникова Н.І

Львів-2019 р.

Лабораторна робота № 5

Тема: Знаходження найкоротшого маршруту за алгоритмом Дейкстри. Плоскі планарні графи.

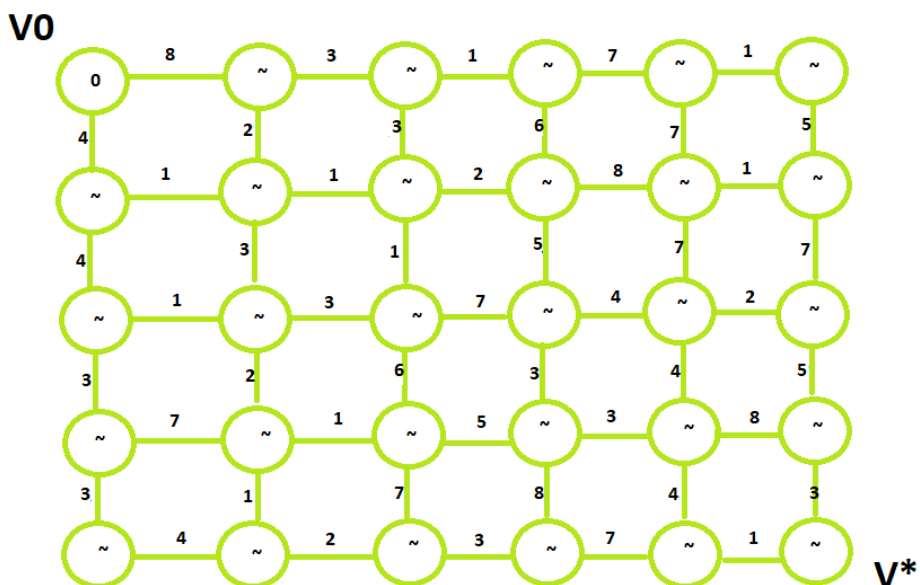
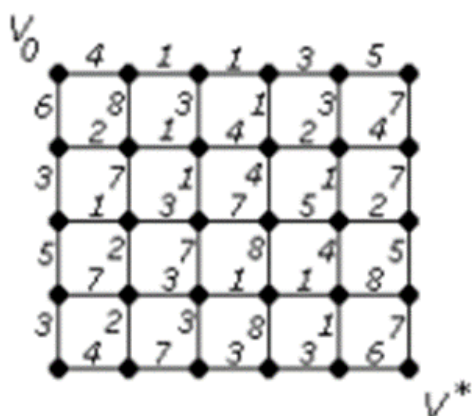
Мета роботи: набуття практичних вмінь та навичок з використання алгоритму Дейкстри.

Варіант 13

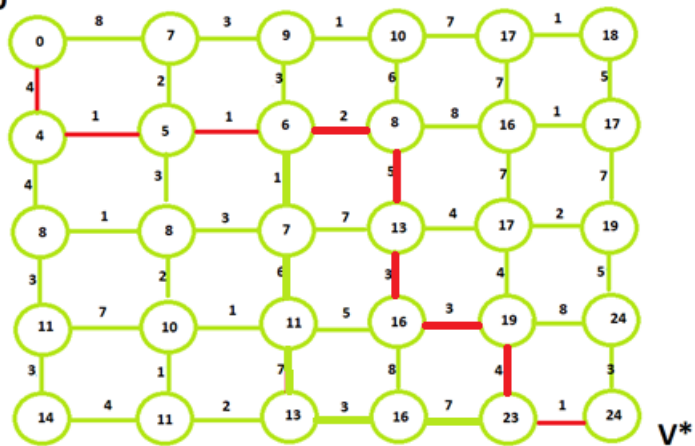
Завдання 1

Розв'язати на графах наступні 2 задачі:

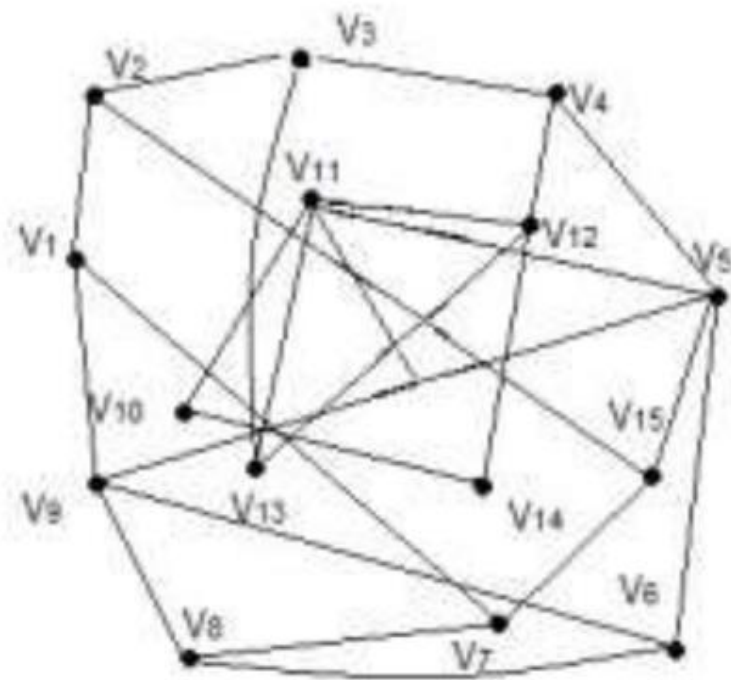
1. За допомогою алгоритму Дейкстра знайти найкоротший шлях у графі поміж парою вершин V_0 і V^* .

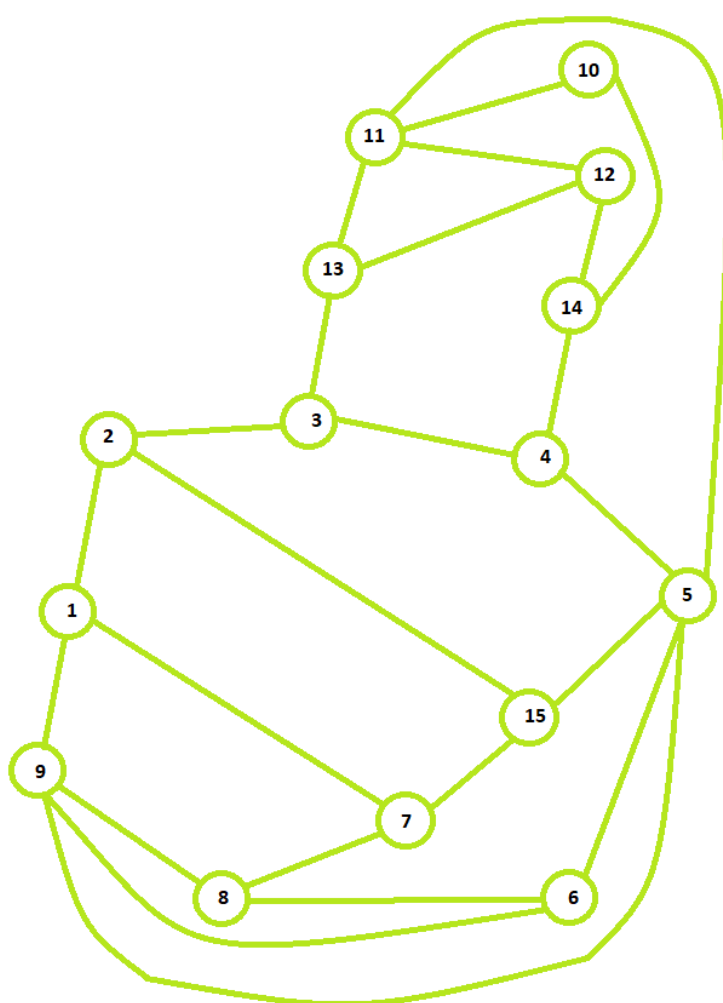
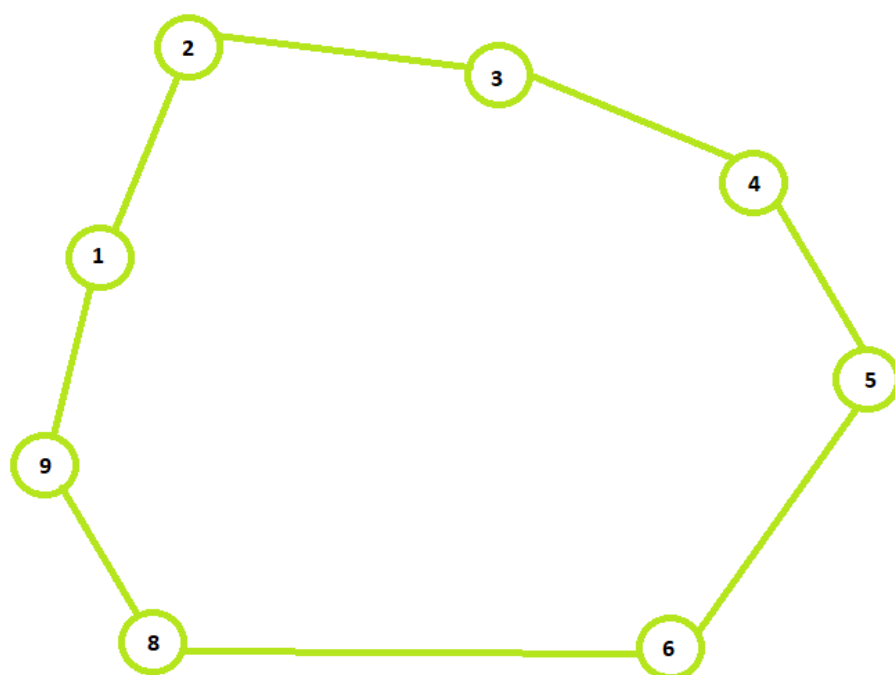


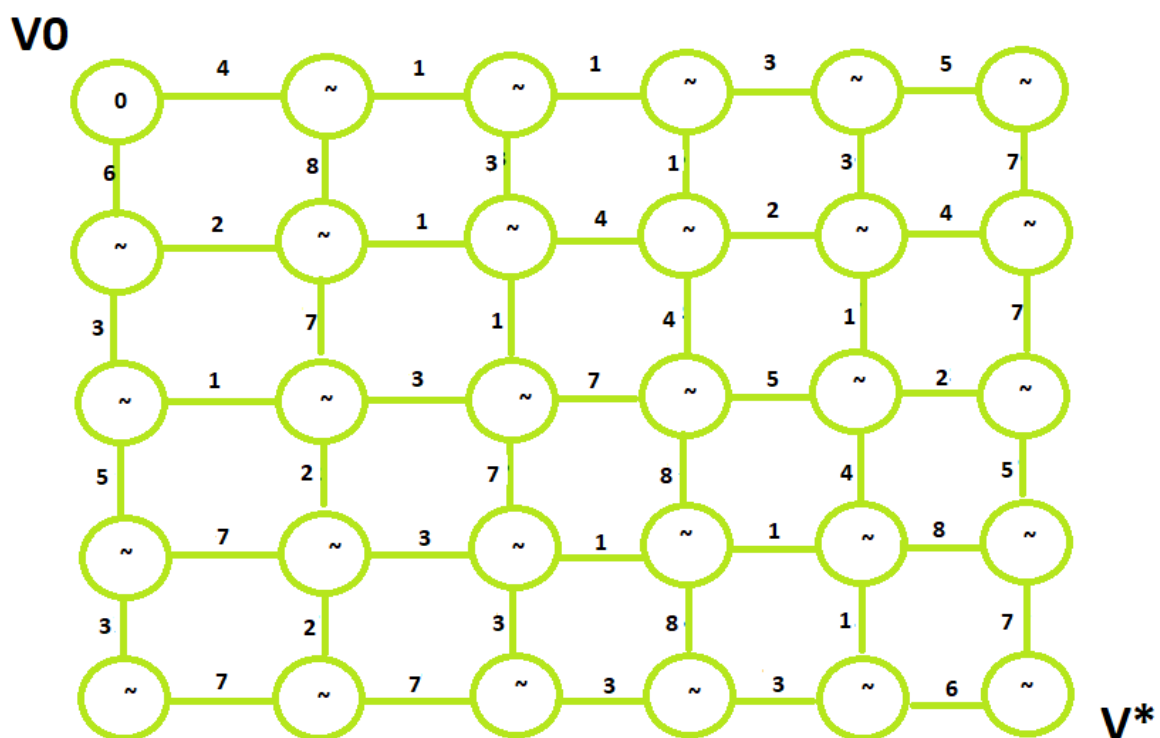
V_0



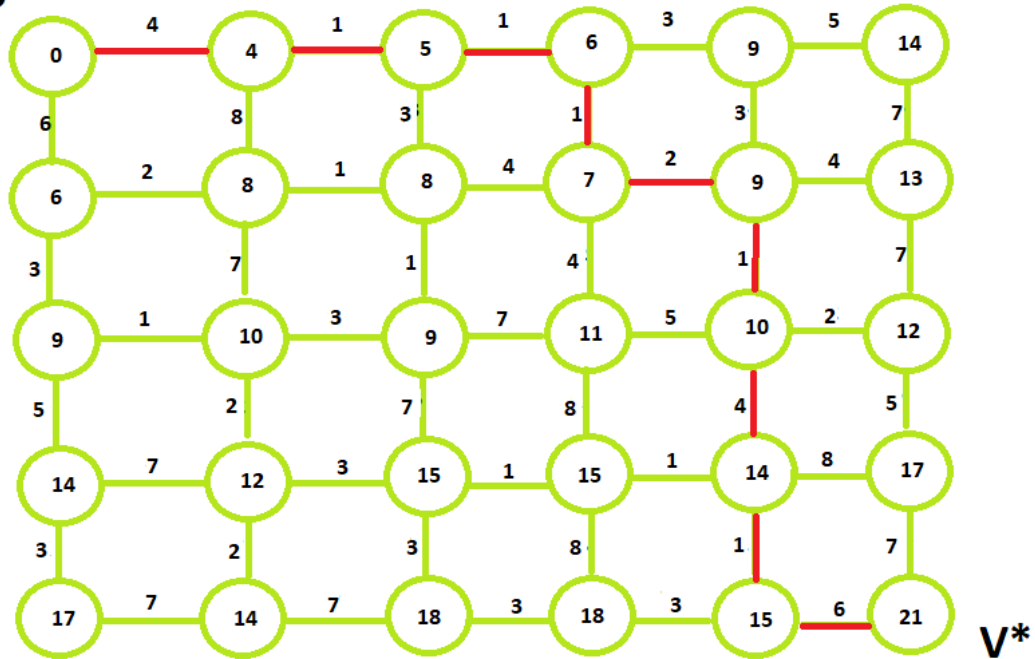
2. За допомогою у -алгоритма зробити укладку графа у площині, або довести що вона неможлива.







V0



```
#define _CRT_SECURE_NO_WARNINGS
#include<iostream>
using namespace std;
int main()
{
    int maximalNumber = 10000;
    int vertex ;//к-сть вершин
    cout << "Amount of vertexes: ";
    cin >> vertex;
    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] = 4;
    a[0][6] = 6;
    a[1][0] = 4;
    a[1][2] = 1;
    a[1][7] = 8;
    a[2][1] = 1;
    a[2][8] = 3;
    a[2][3] = 1;
    a[3][2] = 1;
```

```
a[3][4] = 3;
a[3][9] = 1;
a[4][3] = 3;
a[4][5] = 5;
a[4][10] = 5;
a[5][4] = 5;
a[5][11] = 7;
a[6][0] = 6;
a[6][7] = 2;
a[6][12] = 3;
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] = 3;
a[10][16] = 1;
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] = 2;
a[13][14] = 3;
a[14][13] = 3;
a[14][8] = 1;
a[14][20] = 7;
a[14][15] = 7;
a[15][14] = 7;
a[15][9] = 4;
a[15][21] = 8;
a[15][16] = 5;
a[16][15] = 5;
a[16][10] = 1;
a[16][22] = 4;
a[16][17] = 2;
a[17][16] = 2;
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] = 2;
a[19][25] = 2;
a[19][20] = 3;
a[20][19] = 3;
a[20][14] = 7;
a[20][26] = 3;
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] = 4;
a[22][28] = 1;
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] = 3;
a[26][27] = 3;
a[27][26] = 3;
a[27][21] = 8;
a[27][28] = 3;
a[28][27] = 3;
a[28][22] = 1;
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;
        }
    }
} while (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] << " ";
}
else {
    cout << "There isnt such way";
}
return 0;
}

```

```

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

From vertex : 1
To: 30
Minimal ways to vertex:
0 4 5 6 9 14 6 8 8 7 9 13 9 10 9 11 10 12 14 12 15 15 14 17 17 14 18 18 15 21
Print minimal way
1 2 3 4 10 11 17 23 29 30
Process finished with exit code 0

```

Висновок

Я набула практичних вмінь та навичок з використання алгоритму Дейкстри.