

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ
“ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота №5

з дисципліни

«Дискретна математика»

Виконав:

студент групи КН-112

Хедик Адольф

Викладач:

Мельникова

Н.І.

Львів – 2019р.

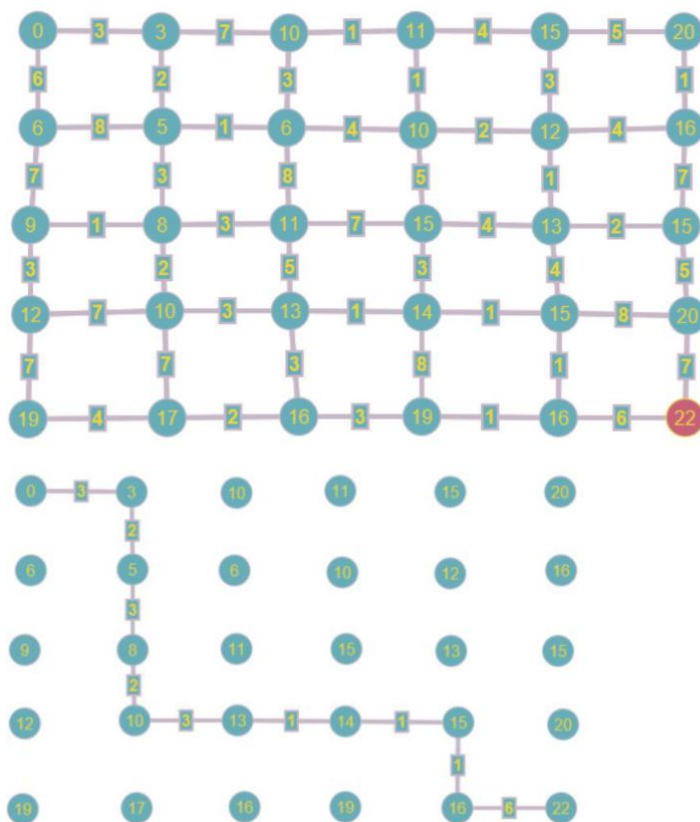
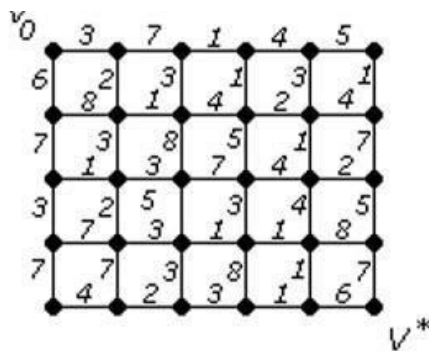
Плоскі планарні графи

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

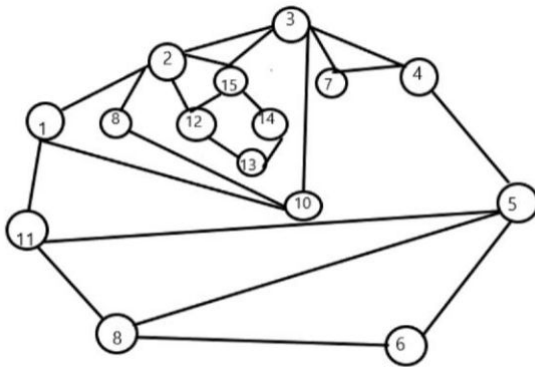
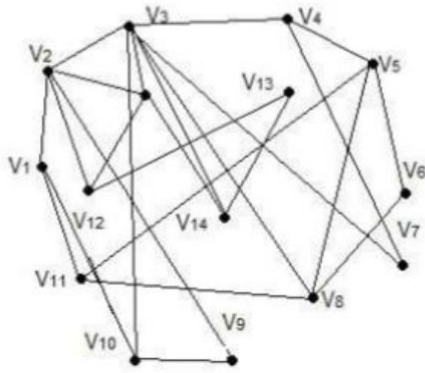
Варіант 16

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^* .



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



Завдання №2. Написати програму, яка реалізує алгоритм Дейкстри знаходження найкоротшого шляху між парою вершин у графі.
Протестувати розроблену програму на графі згідно свого варіанту

Код програми:

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 30
int main(){
    int a[SIZE][SIZE];
    int d[SIZE];
    int v[SIZE];
    int temp, minindex, min;
    int begin_index = 0;
    system("chcp 1251");
    system("cls");
    for (int i = 0; i < 30; i++) {
        for (int j = 0; j < 30; j++) {
            a[i][j] = 0;
        }
    }
}
```

// задання значень ребер (програмно довга)

```
temp = 3;  
a[0][1] = temp;  
a[1][0] = temp;  
temp = 7;  
a[1][2] = temp;  
a[2][1] = temp;  
temp = 1;  
a[3][2] = temp;  
a[2][3] = temp;  
temp = 4;  
a[4][3] = temp;  
a[3][4] = temp;  
temp = 5;  
a[5][4] = temp;  
a[4][5] = temp;  
temp = 8;
```

```
a[7][6] = temp;  
a[6][7] = temp;  
temp = 1;  
a[7][8] = temp;  
a[8][7] = temp;  
temp = 4;  
a[9][8] = temp;  
a[8][9] = temp;  
temp = 2;  
a[9][10] = temp;  
a[10][9] = temp;  
temp = 4;  
a[11][10] = temp;  
a[10][11] = temp;
```

```
temp = 1;  
a[13][12] = temp;  
a[12][13] = temp;  
temp = 3;  
a[14][13] = temp;  
a[13][14] = temp;  
temp = 7;
```

```
a[15][14] = temp;  
a[14][15] = temp;  
temp = 4;  
a[16][15] = temp;  
a[15][16] = temp;  
temp = 2;  
a[16][17] = temp;  
a[17][16] = temp;
```

```
temp = 7;  
a[19][18] = temp;  
a[18][19] = temp;  
temp = 3;  
a[20][19] = temp;  
a[19][20] = temp;  
temp = 1;  
a[21][20] = temp;  
a[20][21] = temp;  
temp = 1;  
a[22][21] = temp;  
a[21][22] = temp;  
temp = 8;  
a[23][22] = temp;  
a[22][23] = temp;
```

```
temp = 4;  
a[25][24] = temp;  
a[24][25] = temp;  
temp = 2;  
a[26][25] = temp;  
a[25][26] = temp;  
temp = 3;  
a[26][27] = temp;  
a[27][26] = temp;  
temp = 1;  
a[27][28] = temp;  
a[28][27] = temp;  
temp = 6;  
a[29][28] = temp;  
a[28][29] = temp;
```

```
temp = 6;  
a[0][6] = temp;  
a[6][0] = temp;  
temp = 2;  
a[1][7] = temp;  
a[7][1] = temp;  
temp = 3;  
a[2][8] = temp;  
a[8][2] = temp;  
temp = 1;  
a[9][3] = temp;  
a[3][9] = temp;  
temp = 3;  
a[4][10] = temp;  
a[10][4] = temp;  
temp = 1;  
a[5][11] = temp;  
a[11][5] = temp;
```

```
temp = 7;  
a[6][12] = temp;  
a[12][6] = temp;  
temp = 3;  
a[7][13] = temp;  
a[13][7] = temp;  
temp = 8;  
a[8][14] = temp;  
a[14][8] = temp;  
temp = 5;  
a[9][15] = temp;  
a[15][9] = temp;  
temp = 1;  
a[10][16] = temp;  
a[16][10] = temp;  
temp = 7;  
a[11][17] = temp;  
a[17][11] = temp;
```

```
temp = 3;
```

```
a[12][18] = temp;
a[18][12] = temp;
temp = 2;
a[13][19] = temp;
a[19][13] = temp;
temp = 5;
a[14][20] = temp;
a[20][14] = temp;
temp = 3;
a[21][15] = temp;
a[15][21] = temp;
temp = 4;
a[16][22] = temp;
a[22][16] = temp;
temp = 5;
a[17][23] = temp;
a[23][17] = temp;
```

```
temp = 7;
a[18][24] = temp;
a[24][18] = temp;
temp = 7;
a[19][25] = temp;
a[25][19] = temp;
temp = 3;
a[20][26] = temp;
a[26][20] = temp;
temp = 8;
a[21][27] = temp;
a[27][21] = temp;
temp = 1;
a[22][28] = temp;
a[28][22] = temp;
temp = 7;
a[23][29] = temp;
a[29][23] = temp;
for (int i = 0; i<SIZE; i++)
{
    for (int j = 0; j<SIZE; j++)
        printf("%5d ", a[i][j]);
```

```

    printf("\n");
}
for (int i = 0; i<SIZE; i++)
{
    d[i] = 10000;
    v[i] = 1;
}
d[begin_index] = 0;
do {
    minindex = 10000;
    min = 10000;
    for (int i = 0; i<SIZE; i++)
    {
        if ((v[i] == 1) && (d[i]<min))
        {
            min = d[i];
            minindex = i;
        }
    }
}

if (minindex != 10000)
{
    for (int i = 0; i<SIZE; i++)
    {
        if (a[minindex][i] > 0)
        {
            temp = min + a[minindex][i];
            if (temp < d[i])
            {
                d[i] = temp;
            }
        }
    }
    v[minindex] = 0;
}
} while (minindex < 10000);

printf("\nsmallest distances: \n");
for (int i = 0; i<SIZE; i++)
    printf("%5d ", d[i]);

```


Результат виконання: