

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

**Кафедра систем штучного інтелекту**



**Лабораторна робота  
з дисципліни  
«Дискретна математика»**

**Виконав:  
студент групи КН-110  
Гавриляк Тарас  
Викладач:  
Мельникова Н.І.**

**Львів – 2018 р.**

## Лабораторна робота № 5.

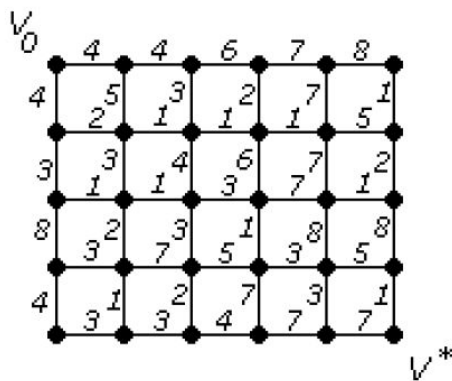
**Тема: Знаходження найкоротшого маршруту за алгоритмом**

**Дейкстри. Плоскі планарні графи**

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

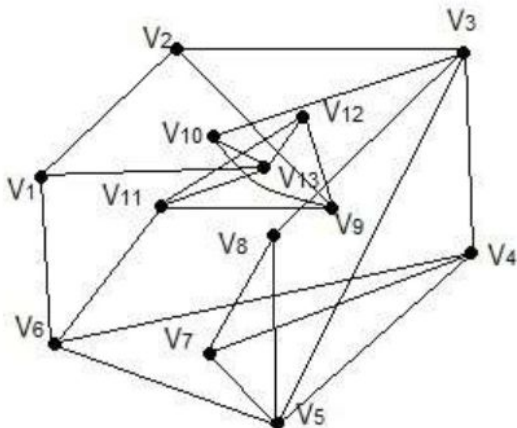
**Завдання № 1. Розв'язати на графах наступні 2 задачі:**

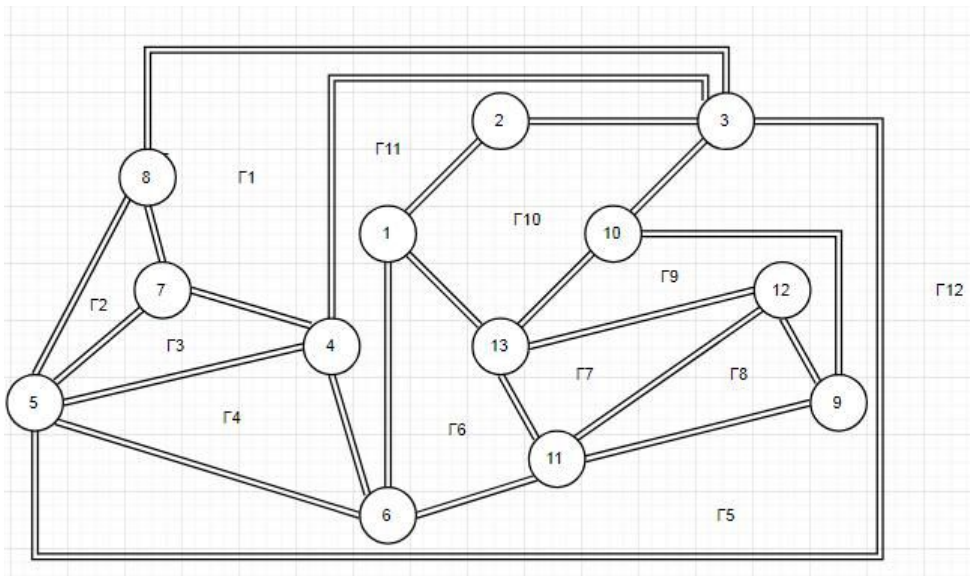
**1. За допомогою алгоритму Дейкстра знайти найкоротший шлях у графі поміж парою вершин  $V_0$  і  $V^*$ .**



Найкоротший шлях з  $V_0 - V^* = 22$

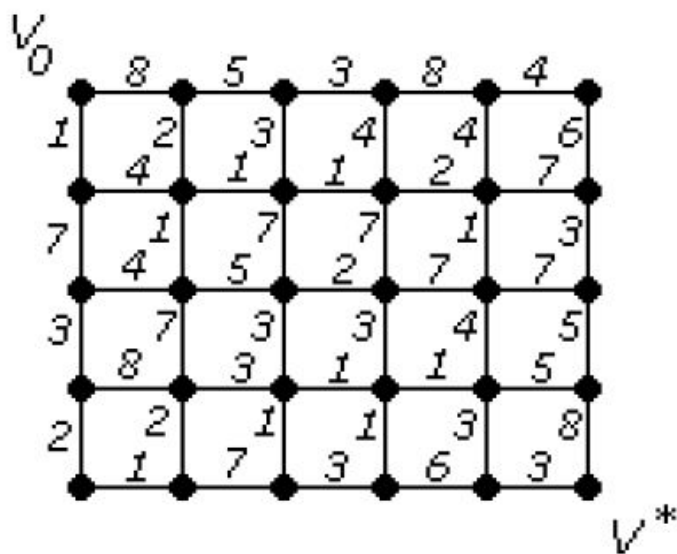
**Завдання №2.** За допомогою -алгоритма зробити укладку графа у площині, або довести що вона неможлива.





Написати програму, яка реалізує алгоритм

Дейкстри знаходження найкоротшого шляху між парою вершин у графі. Протестувати розроблену програму на графі згідно свого варіанту.



Код: #include <stdio.h>

#include <stdlib.h>

#define SIZE 30 // amount of tops- size of matrix

```
#define COMPARE 1000
```

```
int MatrixArray[SIZE][SIZE];    // array of graph  
int Minimaldistance[SIZE];      // array of minimal distance  
int tops[SIZE];                 // array of Tops
```

```
// func to write eight of edges  
void enterEdges();
```

```
// func to initizilation array of distanse and tops  
void initArray();
```

```
// func to output smallest distance to tops (1-30)  
void printDistance();
```

```
int main(void)  
{
```

```
    int temp;        // temp valuable fro distances  
    int minindex;  
    int min;         // temp value of edges
```

```

enterEdges();

initArray();
do
{
    minindex=COMPARE;    // any big value
    min=COMPARE;
    for (int i=0; i<SIZE; i++)
    {
        if ((tops[i]==1) && (Minimaldistance[i]<min))    //if top isnt cheked
and weight < min
        {
            min=Minimaldistance[i];
            minindex=i;
        }
    }

    if(min!=COMPARE)
    {
        for(int i=0; i<SIZE; i++)
        {
            if(MatrixArray[minindex][i]>0)    // if connection with top
            {
                temp = min+ MatrixArray[minindex][i];
                if(temp<Minimaldistance[i])    //diferrent distance to top

```

```

        {
            Minimaldistance[i] = temp;
        }
    }
}

tops[minindex]=0; //top is cheked
}
}

while(minindex < COMPARE);

printDistance();

// way
int ver[SIZE];    // cheked tops
int end = 29;
ver[0] = end + 1;
int k = 1;        // index of previous top
int weight = Minimaldistance[end];
while (end > 0)
{
    for(int i=0; i<SIZE; i++)
        if (MatrixArray[end][i] != 0) // if connecion is
        {
            int temp = weight - MatrixArray[end][i];
            if (temp == Minimaldistance[i])
            {

```

```

        weight = temp;    // saving new weight
        end = i;
        ver[k] = i + 1;
        k++;
    }
}

// output way
printf("\n the smallest way from V0 o V*:\n");
for(int i = k-1; i>=0; i--)
    printf("(v%d) ",ver[i]);
printf("\n");

return 0;
}

void enterEdges()
{
    printf("Enter tops:\n");
    int r, c, n;
    for(int i=0; i<49; i++)
    {
        scanf("%d %d %d", &r, &c, &n);
        MatrixArray[r-1][c-1]=n;
        MatrixArray[c-1][r-1]=n;
    }
}

```

```

}

void initArray()
{
    for(int i=0; i<SIZE; i++)
    {
        Minimaldistance[i]=COMPARE;    // any value for tops untill they aren't
defined
        tops[i]=1;    // aren't cheking tops = 1
    }
    Minimaldistance[0]= 0;    // distance to 1 tops (0)
}

void printDistance()
{
    printf("\nSmallest way to every top(1-30): \n");
    for(int i=0; i<SIZE; i++)
    {
        printf("%d) %d; ", i+1, Minimaldistance[i]);

        printf("\n");
    }
}

```

**Висновок:** Навчився на практиці застосовувати алгоритм дейкстри.



