

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

*Кафедра систем штучного інтелекту*

**Звіт**  
*Лабораторна робота №5*  
з дисципліни:  
“Дискретна математика”

**Виконав:**

Студент групи КН-113

Вовчак Л. В.

**Викладач:**

Мельникова Н.І.

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

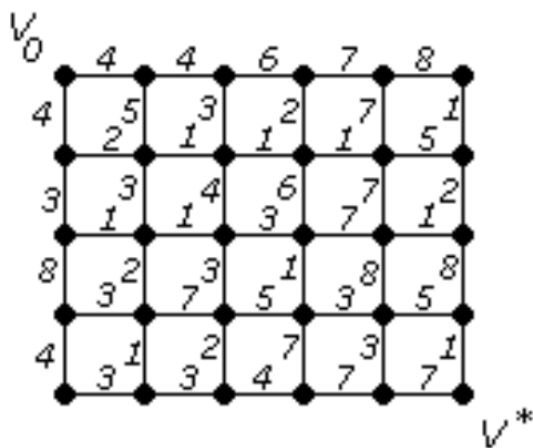
**Мета:** набуття практичних вмінь та навичок з використання алгоритму Дейкстри.

**Постановка завдання:**

**Варіант 6**

**Завдання 1.**

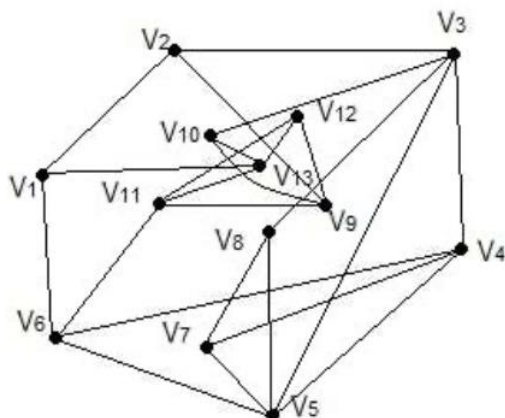
1. За допомогою алгоритму Дейкстри знайти найкоротший шлях \* у графі поміж парою вершин  $V_0$  і  $V^*$ .



**Завдання 2.**

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

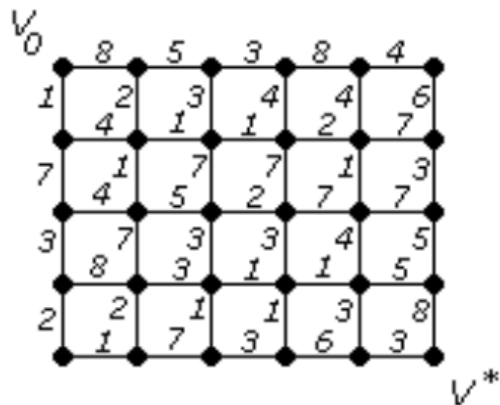
**6**



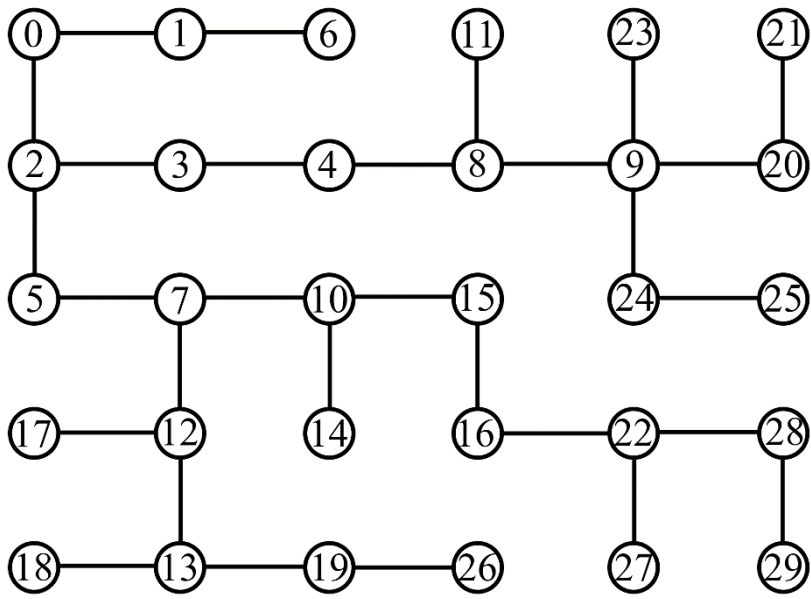
## Додаток 2

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

6



## Розв'язок задачі №1



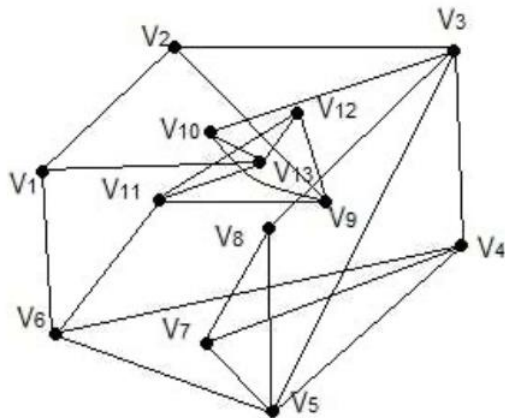
Найкоротший шлях з вершини  $V(0)$  до вершини  $V^*(29)$  пролягає через вершини 0-2-5-7-10-15-16-22-28-29 і дорівнює 22 одиниці виміру.

Відстані до всіх інших вершин від вершини  $V(0)$ :

L1 = 4;	L11 = 10;	L21 = 15;
L2 = 4;	L12 = 10;	L22 = 16;
L3 = 6;	L13 = 11;	L23 = 16;
L4 = 7;	L14 = 12;	L24 = 16;
L5 = 7;	L15 = 12;	L25 = 17;
L6 = 8;	L16 = 13;	L26 = 18;
L7 = 8;	L17 = 13;	L27 = 19;
L8 = 8;	L18 = 14;	L28 = 21;
L9 = 9;	L19 = 14;	L29 = 22;
L10 = 9;	L20 = 14;	L0 = 0;

## Розв'язок задачі №2

6



Необхідною умовою доведення непланарності графа є наявність більше п'ятьох і більше вершин зі степенем більше 3 або шістьох і більше вершин зі степенем більше 2.

На даному графі можемо зауважити, як мінімум 5 вершин зі степенем більше 3, зокрема вершини 3, 4, 5, 9 та 13.

Отже граф непланарний та здійснити  $\gamma$ -укладку графа на площині неможливо.

## Розв'язок додатку №2

## Код програмної реалізації

```
#include <iostream>
```

```
#include <bits/stdc++.h>
```

```
#define size 30
```

```
using namespace std;
```

```
int main()
```

 $\{$ 

```
setlocale(LC_ALL, "Ukrainian");
```

```
int A[size][size] =
```

[illegible]

```
};
```

```
int d[size], v[size], t, min_i, min, beg_i(0);
```

```
for (int i = 0; i < size; i++)
```

```
{
```

```
    d[i] = 10000;
```

```
    v[i] = 1;
```

```
}
```

```
d[beg_i] = 0;
```

```
do {
```

```
    min_i = 10000;
```

```
    min = 10000;
```

```
    for (int i = 0; i < size; i++)
```

```
    {
```

```
        if ((v[i] == 1) && (d[i] < min))
```

```
        {
```

```
            min = d[i];
```

```
            min_i = i;
```

```
        }
```

```
    }
```

```
if (min_i != 10000)
```

```
{
```

```
    for (int i = 0; i < size; i++)
```

```
    {
```

```
        if (A[min_i][i] > 0)
```

```
        {
```

```
            t = min + A[min_i][i];
```

```
            if (t < d[i])
```

```
            {
```

```
                d[i] = t;
```

```
            }
```

```
        }
```

```
    }
```

```
    v[min_i] = 0;
```

```
}
```

```
} while (min_i < 10000);
```

```

    cout<<"Найкоротша відстань від вершини V0 до кожної з вершин:
"<<endl;
    for (int i = 0; i<size; i++)
        cout<<"V"<<i+1<<" - "<<d[i]<<endl;

    int ver[size];
    int end = 29;
    ver[0] = end + 1;
    int k = 1;
    int weight = d[end];

    while (end != beg_i)
    {
        for (int i = 0; i<size; i++){
            if (A[end][i] != 0)
            {
                int t = weight - A[end][i];
                if (t == d[i])
                {
                    weight = t;
                    end = i;
                    ver[k] = i + 1;
                    k++;
                }
            }
        }
    }

    cout<<"Вивід найкоротшого шляху: "<<endl;
    for (int i = k - 1; i >= 0; i--){
        if(ver[i]!=size)
            cout<<"V"<<ver[i]<<" -> ";
        else
            cout<<"V"<<ver[i];
    }
    cout<<endl;
    return 0;
}

```



## Результат програми

```
Найкоротша відстань від вершини V0 до кожної з вершин:  
V1 - 0  
V2 - 7  
V3 - 9  
V4 - 11  
V5 - 13  
V6 - 17  
V7 - 1  
V8 - 5  
V9 - 6  
V10 - 7  
V11 - 9  
V12 - 16  
V13 - 8  
V14 - 6  
V15 - 11  
V16 - 13  
V17 - 10  
V18 - 17  
V19 - 11  
V20 - 13  
V21 - 14  
V22 - 15  
V23 - 14  
V24 - 19  
V25 - 13  
V26 - 14  
V27 - 15  
V28 - 16  
V29 - 17  
V30 - 20  
Вивід найкоротшого шляху:  
V1 -> V7 -> V8 -> V9 -> V10 -> V11 -> V17 -> V23 -> V29 -> V30  
  
Process returned 0 (0x0)   execution time : 0.135 s  
Press any key to continue.  
_
```

**Висновок:** Виконуючи дану лабораторну роботу, я отримав навички в роботі з знаходженням найкоротшого маршруту за алгоритмом Дейкстри.