

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

Кафедра систем штучного інтелекту

Лабораторна робота
з дисципліни
«Дискретна математика»

Виконав:
студент групи КН-109
Кіндрат
Володимир
Викладач:
Мельникова Н.І.

Львів – 2018 р.

Лабораторна робота № 5.

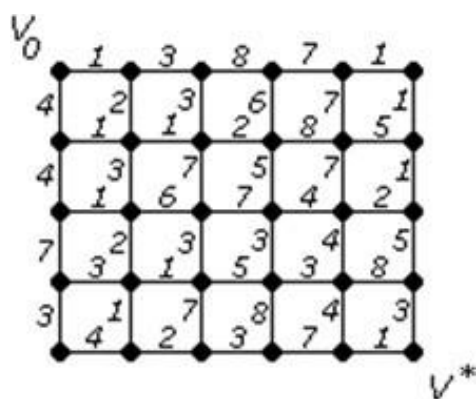
Тема: Знаходження найкоротшого маршруту за алгоритмом Дейкстри. Плоскі планарні графи

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

Варіант № 5

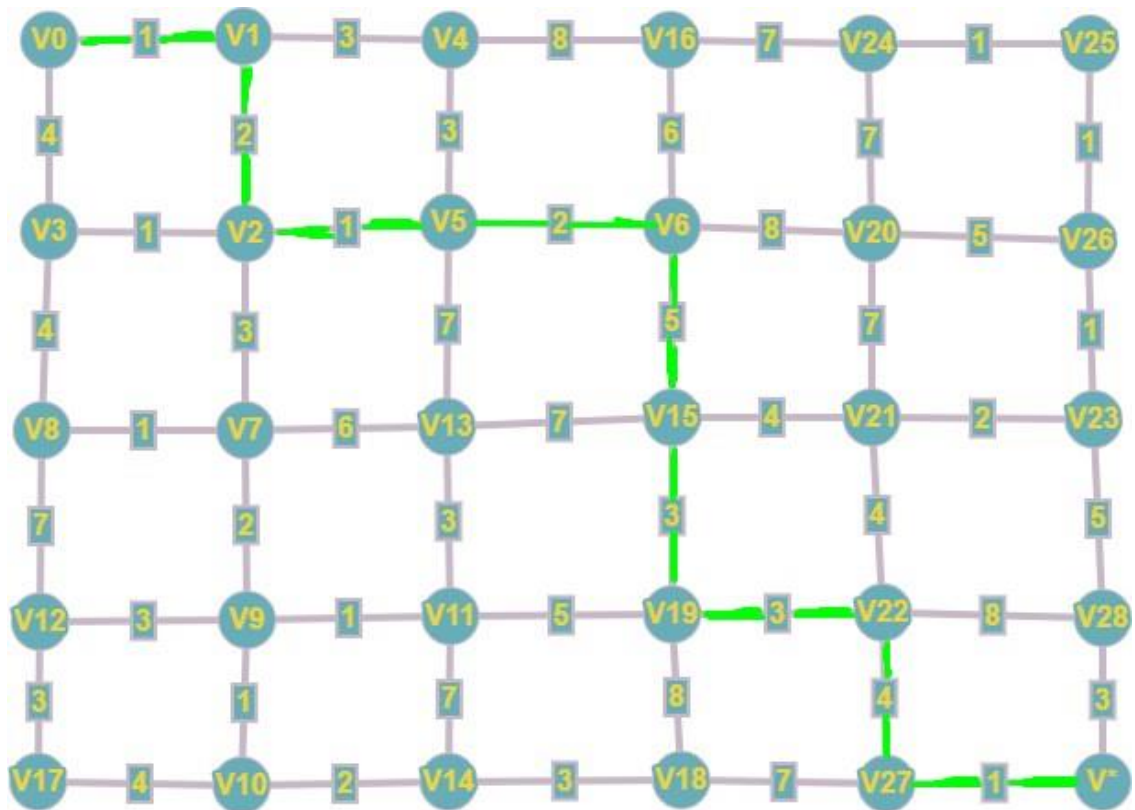
Завдання № 1. Розв'язати на графах наступні 2 задачі:

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

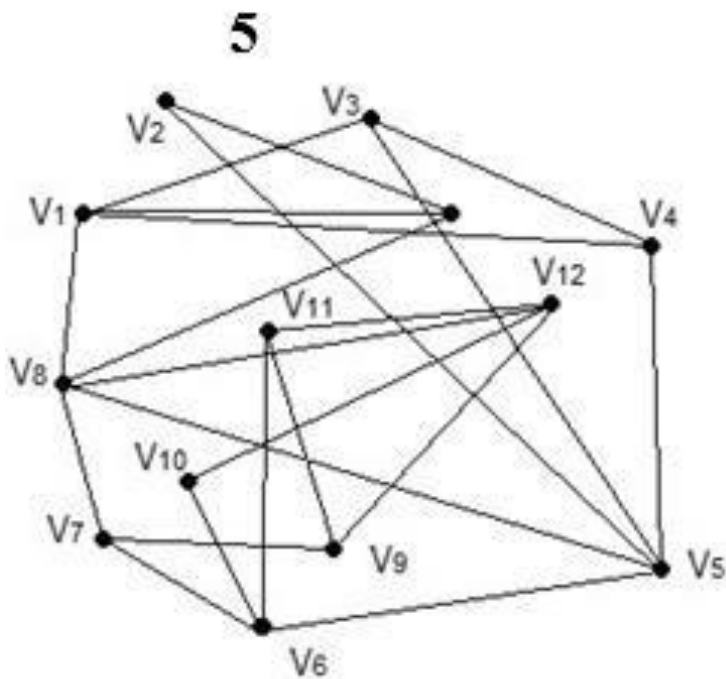


$L(V_0)=0$	$L(V_{10})=9$	$L(V_{20})=14$
$L(V_1)=1$	$L(V_{11})=9$	$L(V_{21})=15$
$L(V_2)=3$	$L(V_{12})=11$	$L(V_{22})=17$
$L(V_3)=4$	$L(V_{13})=11$	$L(V_{23})=17$
$L(V_4)=4$	$L(V_{14})=11$	$L(V_{24})=19$
$L(V_5)=4$	$L(V_{15})=11$	$L(V_{25})=20$
$L(V_6)=6$	$L(V_{16})=12$	$L(V_{26})=21$
$L(V_7)=6$	$L(V_{17})=13$	$L(V_{27})=21$
$L(V_8)=7$	$L(V_{18})=14$	$L(V_{28})=22$
$L(V_9)=8$	$L(V_{19})=14$	$L(V^*)=22$

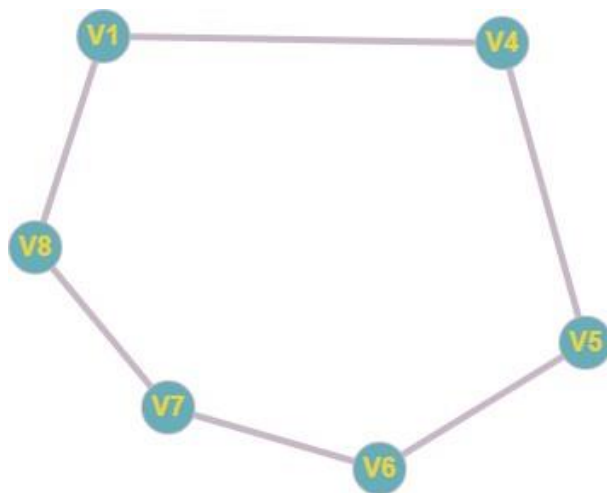
Найкоротший шлях від вершини V_0 до вершини V^* :
 $V_0 - V_1 - V_2 - V_5 - V_6 - V_{15} - V_{19} - V_{22} - V_{27} - V^*$



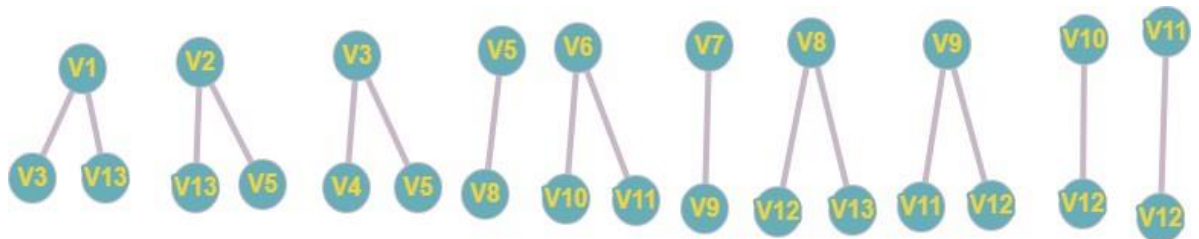
2. За допомогою γ -алгоритма зробити укладку графа у площині, або довести що вона неможлива.



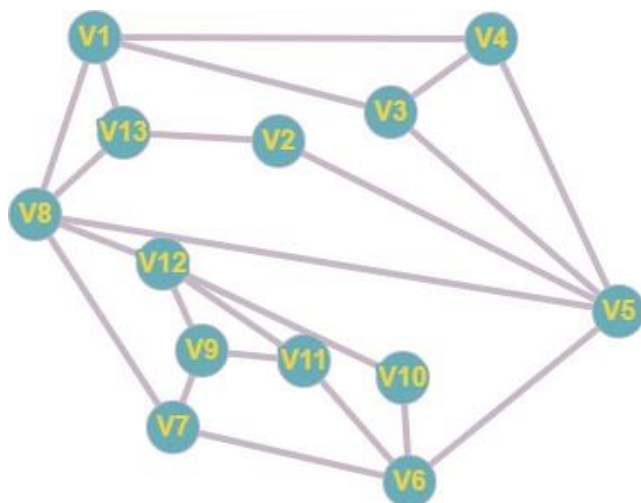
Цикл:



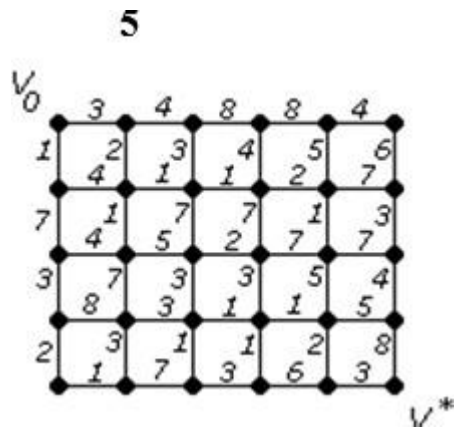
Сегменты:



Плоский планарный граф:



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 6
int main()
{
    int a[SIZE][SIZE];
    int d[SIZE];
    int v[SIZE];
    int temp;
    int minindex, min;
    system("chcp 1251");
    system("cls");
    for (int i = 0; i < SIZE; i++)
    {
        a[i][i] = 0;
        for (int j = i + 1; j < SIZE; j++) {
            printf("Введіть відстань %d - %d: ", i + 1, j + 1);
            scanf("%d", &temp);
            a[i][j] = temp;
            a[j][i] = 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[0] = 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("\nНайкоротші відстані до вершин \n");
    for (int i = 0; i < SIZE; i++)
        printf("%5d ", d[i]);

    int ver[SIZE];
    int end = 4;
    ver[0] = end + 1;
    int k = 1;
    int weight = d[end];

    while (end > 0)
    {
        for (int i = 0; i < SIZE; i++)
            if (a[end][i] != 0)
            {
                temp = weight - a[end][i];
                if (temp == d[i])

```

```
        {
            weight = temp;
            end = i;
            ver[k] = i + 1;
            k++;
        }
    }

    printf("\nНайкоротший шлях\n");
    for (int i = k - 1; i >= 0; i--)
        printf("%3d ", ver[i]);
    getchar(); getchar();
    return 0;
}
```