

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

Кафедра систем штучного інтелекту

Лабораторна робота №5

з дисципліни
«Дискретна математика»

Виконав:
студент групи КН-109
Гречух Тарас
Викладач:
Мельникова Н.І.

Львів – 2018 р.

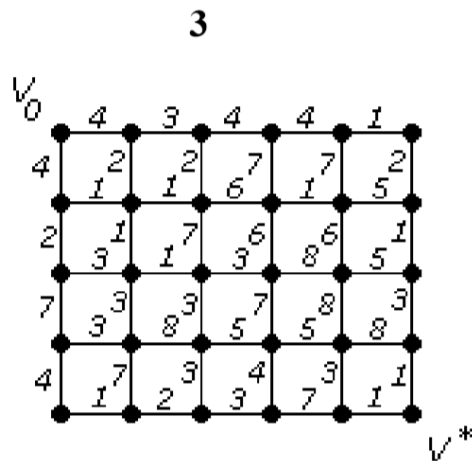
Тема: Знаходження найкоротшого маршруту за алгоритмом Дейкстри. Плоскі планарні графи.

Мета: Набуття практичних вмінь та навичок з використання алгоритму Дейкстри.

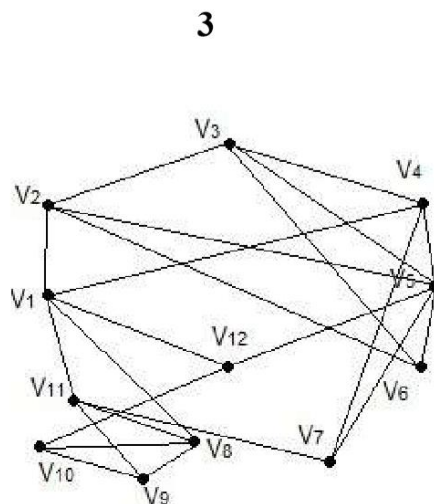
Завдання з додатку 1 (варіант 3):

Розв'язати на графах наступні 2 задачі:

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

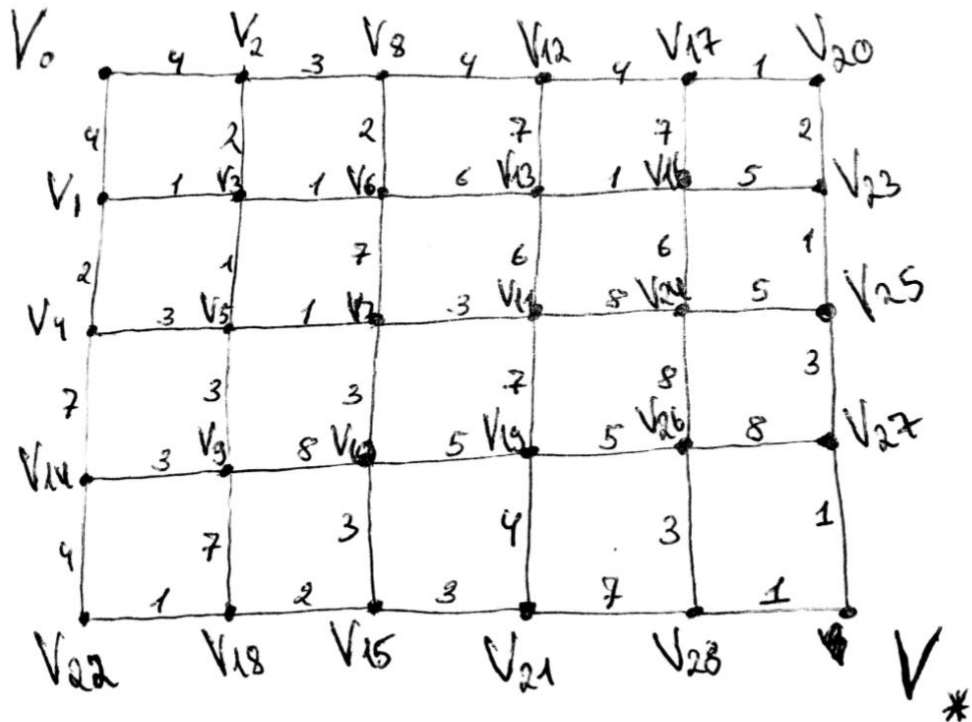


2. За допомогою у-алгоритма зробити укладку графа у площині, або довести що вона неможлива.



Розв'язання:

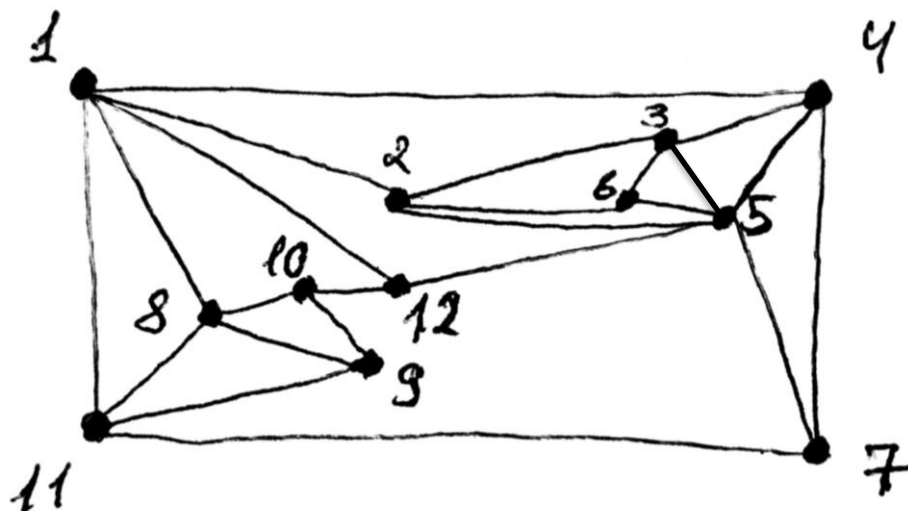
1.



Поступово знаходимо найкоротші шляхи (якщо з'явився коротший – заміняємо старий на новий). Відповідно нумеруємо вершини по зростанню ваги. На рисунку зображене поступове нумерування вершин. Найкоротший шлях отримали: $(V_0 - V_2 - V_8 - V_{12} - V_{17} - V_{20} - V_{23} - V_{25} - V_{27} - V^*) \Rightarrow (4 + 3 + 4 + 4 + 1 + 2 + 1 + 3 + 1) = 23 \Rightarrow$ його вага дорівнює 23.

В: 23.

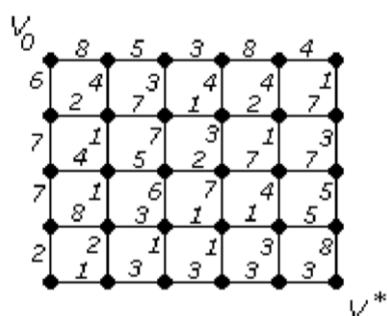
2.



Послідовність дій: Спершу в даному умовою графі обрався довільний цикл, а саме цикл ($V_1 - V_2 - V_3 - V_4 - V_5 - V_7 - V_{11}$). Потім створювалися сегменти уже задіяних вершин та сегменти відсутніх вершин. Сегменти рисувалися з усіма варіантами з'єднань, а потім повторні ребра вилучалися. Почав я із з'єднання уже задіяних вершин, а саме (1-4), (3-5), (2-5) та (4-7). Пізніше почав залучати сегменти відсутніх вершин, користуючись алгоритмом (врахування у якому секторі лежать вершини). Граф уклався у площині, отже він плоский, ребра його не перетинаються, отже граф планарний!

Завдання з додатку 2 (варіант 3):

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



Розв'язання:

```
#include <iostream>
#include <stdio.h>

using namespace std;

const int V = 30; //розмірність графа

void Dijkstra(int GR[V][V], int st) //Алгоритм Дейкстри
{
    int distance[V]; //масив знайдених найкоротших шляхів
    // допоміжні змінні

    int count;
    int index;
    int i;
    int u;
    int m = st + 1;

    bool visited[V]; //мвсив відвіданих вершин
```

```
for (i = 0; i < V; i++)
{
    distance[i] = INT_MAX; //
    visited[i] = false;
}

distance[st] = 0;

for (count = 0; count < V - 1; count++)
{
    int min = INT_MAX;

    for (i = 0; i < V; i++)
    {
        if (!visited[i] && distance[i] <= min)
        {
            min = distance[i];
            index = i;
        }
    }

    u = index;

    visited[u] = true;

    for (i = 0; i < V; i++)
    {
        if (!visited[i] && GR[u][i] && distance[u] != INT_MAX && distance[u] +
GR[u][i] < distance[i])
        {
            distance[i] = distance[u] + GR[u][i];
        }
    }
}

cout << "Path value for all points:\n";

for (i = 0; i < V; i++)
{
    if (distance[i] != INT_MAX)
    {
        cout << m << " > " << i + 1 << " = " << distance[i] << endl;
    }
    else
    {
        cout << m << " > " << i + 1 << " = " << "Path unavailible" << endl;
    }
}
}

void main()
{
    int start, GR[V][V] = {

        {0,8,0,0,0,0,6,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
        {0,0,5,0,0,0,0,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
        {0,0,0,3,0,0,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
        {0,0,0,0,8,0,0,0,0,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
        {0,0,0,0,0,4,0,0,0,0,4,0,0,0,0,0,0,0,0,0,0,0,0,0},
        {0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0},
        {0,0,0,0,0,0,0,0,2,0,0,0,0,7,0,0,0,0,0,0,0,0,0,0},
        {0,0,0,0,0,0,0,0,0,7,0,0,0,0,1,0,0,0,0,0,0,0,0,0},
        {0,0,0,0,0,0,0,0,0,1,0,0,0,0,7,0,0,0,0,0,0,0,0,0},
        {0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,3,0,0,0,0,0,0,0,0},
        {0,0,0,0,0,0,0,0,0,0,0,7,0,0,0,0,1,0,0,0,0,0,0,0},
        {0,0,0,0,0,0,0,0,0,0,0,0,7,0,0,0,0,0,1,0,0,0,0,0},
        {0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,0,0,0,0,0,0,0,0},
    }
```

```

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,4,0,0,0,0,7,0,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,5,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,6,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,7,0,0,0,0,7,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,7,0,0,0,0,4,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,8,0,0,0,2,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,0,2,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5,0,0,0,3,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,8},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0},
{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},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
};

cout << "Starting point: >> "; cin >> start;

Dijkstra(GR, start - 1);

system("pause");
}

```

Результат виконаної програми:

```

C:\Users\LENOVO\source\repos\LAB5(disc)\Debug\LAB5(disc).exe
Starting point: >> 1
Path value for all points:
1 > 1 = 0
1 > 2 = 8
1 > 3 = 13
1 > 4 = 16
1 > 5 = 24
1 > 6 = 28
1 > 7 = 6
1 > 8 = 8
1 > 9 = 15
1 > 10 = 16
1 > 11 = 18
1 > 12 = 25
1 > 13 = 13
1 > 14 = 9
1 > 15 = 14
1 > 16 = 16
1 > 17 = 19
1 > 18 = 26
1 > 19 = 20
1 > 20 = 10
1 > 21 = 13
1 > 22 = 14
1 > 23 = 15
1 > 24 = 20
1 > 25 = 22
1 > 26 = 12
1 > 27 = 14
1 > 28 = 15

```

```
C:\Users\LENOVO\source\repos\LAB5(disc)\Debug\LAB5(disc).exe
1 > 5 = 24
1 > 6 = 28
1 > 7 = 6
1 > 8 = 8
1 > 9 = 15
1 > 10 = 16
1 > 11 = 18
1 > 12 = 25
1 > 13 = 13
1 > 14 = 9
1 > 15 = 14
1 > 16 = 16
1 > 17 = 19
1 > 18 = 26
1 > 19 = 20
1 > 20 = 10
1 > 21 = 13
1 > 22 = 14
1 > 23 = 15
1 > 24 = 20
1 > 25 = 22
1 > 26 = 12
1 > 27 = 14
1 > 28 = 15
1 > 29 = 18
1 > 30 = 21
Press any key to continue . . .
```

Висновок: Я набув практичних вмінь та навичок з використання алгоритму Дейкстри.