

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
“ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

Лабораторна робота № 5

з дисципліни

«Дискретна математика»

Виконав:

студент групи КН-112

Калітовський Роман

Викладач:

Мельникова Н.І.

Львів – 2019 р.

Лабораторна робота № 5.

Тема: Знаходження найкоротшого маршруту за алгоритмом Дейкстри.

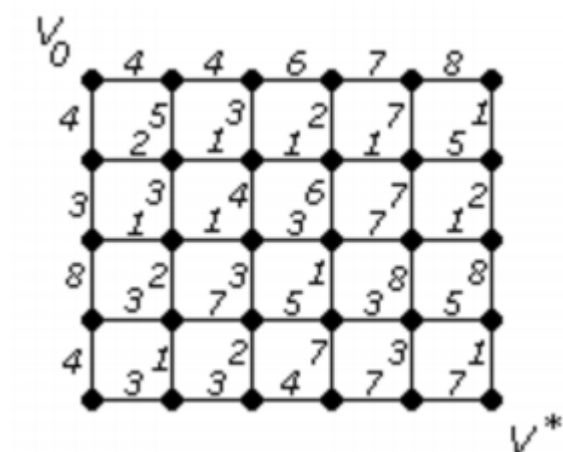
Плоскі планарні графи

Мета: Знаходження найкоротшого маршруту за алгоритмом Дейкстри.

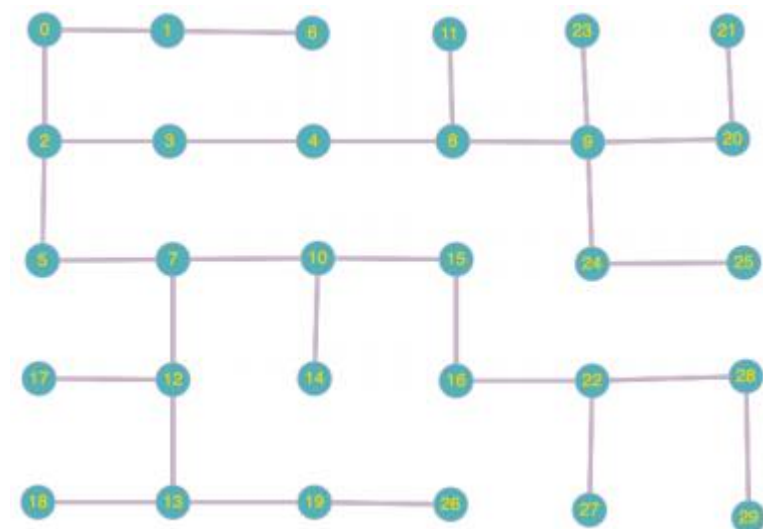
Плоскі планарні графи

Варіант 6

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



Розв'язання:



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.

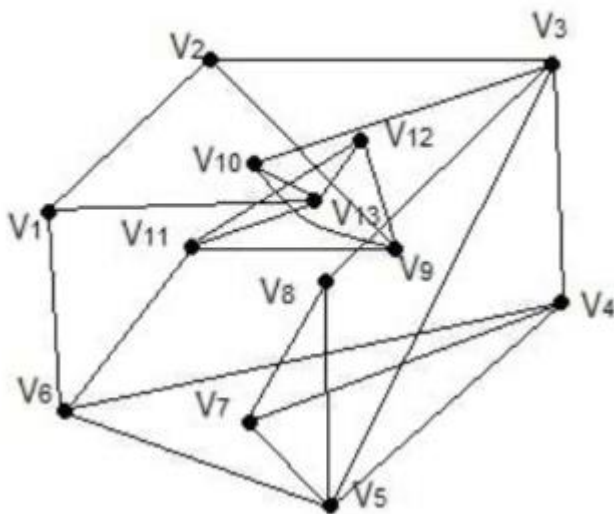
Відстані до всіх інших вершин від вершини $V(0)$:

$L_1 = 4$; $L_{11} = 10$; $L_{21} = 15$;

$L_2 = 4;$	$L_{12} = 10;$	$L_{22} = 16;$
$L_3 = 6;$	$L_{13} = 11;$	$L_{23} = 16;$
$L_4 = 7;$	$L_{14} = 12;$	$L_{24} = 16;$
$L_5 = 7;$	$L_{15} = 12;$	$L_{25} = 17;$
$L_6 = 8;$	$L_{16} = 13;$	$L_{26} = 18;$
$L_7 = 8;$	$L_{17} = 13;$	$L_{27} = 19;$
$L_8 = 8;$	$L_{18} = 14;$	$L_{28} = 21;$
$L_9 = 9;$	$L_{19} = 14;$	$L_{29} = 22;$
$L_{10} = 9;$	$L_{20} = 14;$	$L_0 = 0;$

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

6



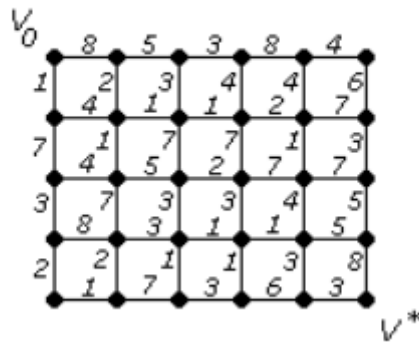
Розв'язання:

Необхідною умовою доведення непланарності графа є наявність більше п'яťох 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 g -укладку графа на площині неможливо.

Завдання №2.

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

6



Програмна реалізація:

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

```
#include <stdlib.h>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int min_top(int** arr, int v) {
```

```
    int m = 0;
```

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

```
    {
```

```
        if (arr[i][1])
```

```
        {
```

```
            m = i;
```

```
            break;
```

```
        }
```

```
    }
```

```
    for (int i = 1; i < v; i++)
```

```
    {
```

```
        if (arr[m][0] >= arr[i][0] && arr[i][1] == 1)
```

```
        {
```

```
            m = i;
```

```
        }
```

```
    }
```

```
    return m;
```

```
}
```

```
int main()
```

```
{
```

```
    int inf = 100000;
```

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

```
    int a, b, c;
```

```
    int v = 0;
```

```
    cout << "Кількість вершин графа : ";
```

```
    cin >> v;
```

```
    int** graph = new int* [v];
```

```
    for (int j = 0; j < v; j++)
```

```

{
    graph[j] = new int[v];
}
for (int a = 0; a < v; a++)
{
    for (int j = 0; j < v; j++)
    {
        graph[a][j] = 0;
    }
}
int r = 0;
cout << "Кількість ребер графа : ";
cin >> r;
cout << "Введіть вагу ребер графа : " << endl;
for (size_t i = 0; i < r; i++)
{
    cin >> a;
    cin >> b;
    cin >> c;
    graph[a - 1][b - 1] = graph[b - 1][a - 1] = c;
}
int p;
int** tops = new int* [v];
for (int j = 0; j < v; j++)
{
    tops[j] = new int[2];
}
int* tops_path = new int[v];
cout << "Вихідна вершина: ";
cin >> p;
for (int i = 0; i < v; i++)
{
    if (i == p - 1) {
        tops[i][0] = 0;
        tops[i][1] = 1;
    }
    else {
        tops[i][0] = inf;
        tops[i][1] = 1;
    }
}
tops_path[p - 1] = 0;
int m;
for (int i = 0; i < v; i++)
{
    m = min_top(tops, v);
    for (int j = 0; j < v; j++)
    {

```

```

        if (graph[m][j])
        {
            if (tops[j][0] > tops[m][0] + (graph[m][j]))
            {
                tops[j][0] = tops[m][0] + (graph[m][j]);
                tops_path[j] = m;
            }
        }
    }
    tops[m][1] = 0;
}
///  

cout << "Введіть потрібну вершину: ";
int k;
cin >> k;
cout << "Мінімальний шлях: ";
cout << tops[k - 1][0];
cout << endl << k << " <-- ";
k--;
for (int a = 0; tops_path[k] != p - 1; a++)
{
    cout << tops_path[k] + 1 << " <-- ";
    k = tops_path[k];
}
cout << p << endl;
return 0;
}

```

Приклад роботи програми:

```

Кількість вершин графа : 30
Кількість ребер графа : 49
Введіть вагу ребер графа :
1 2 8
1 7 1
2 3 5
2 8 2
3 4 3
3 9 3
4 5 8
4 10 4
5 6 4
5 11 4
6 12 6
7 8 4
7 13 7
8 9 1
8 14 1
9 10 1
9 15 7
10 11 2
10 16 7
11 12 7
11 17 1
12 18 3
13 14 4
13 19 3
14 15 5
14 20 7
15 16 2
15 21 3
16 17 7
16 22 3
17 18 7
17 23 4
18 24 5
19 20 8
19 25 2
20 21 3
20 26 2
21 22 1
21 27 1
22 23 1
22 28 1
23 24 5
23 29 3
24 30 8
25 26 1
26 27 7
27 28 3
28 29 6
29 30 3
Вихідна вершина: 1
Введіть потрібну вершину: 30
Мінімальний шлях: 20
30 <-- 29 <-- 23 <-- 17 <-- 11 <-- 10 <-- 9 <-- 8 <-- 7 <-- 1

C:\Users\User\Documents\Дискретна математика\Лабораторні роботи\laba5\Debug\laba5.exe (процесс 2452) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...

```

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