

Кафедра систем штучного інтелекту

## **Звіт**

### **Лабараторна робота № 5**

З дисципліни

**«Дискретна математика»**

**Виконав:**

Студент групи КН-112

Бенчарський Максим

**Викладач:**

Мельникова Н. І.

Львів-2019 р.

**Тема:** “Знаходження найкоротшого маршруту за алгоритмом Дейкстри. Плоскі планарні граfi”

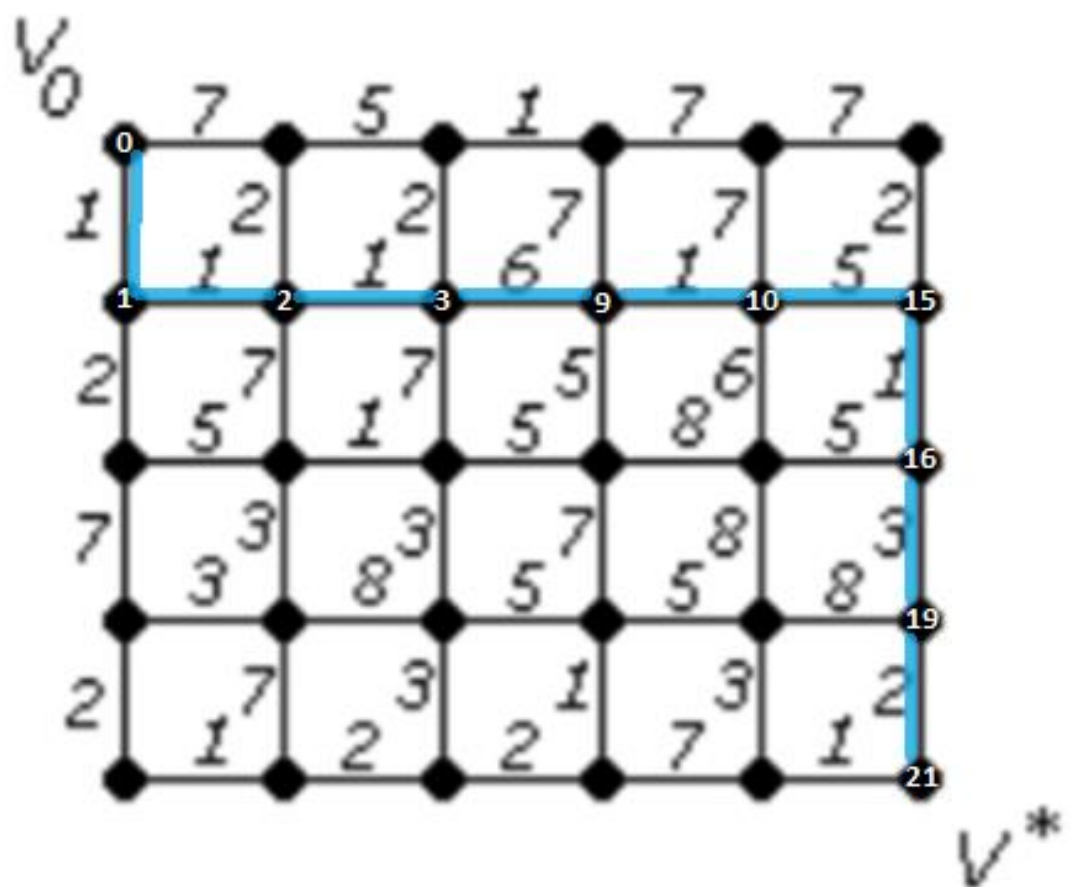
**Мета роботи:** набуття практичних вмінь та навичок з використання алгоритму Дейкстри.

## Варіант №2

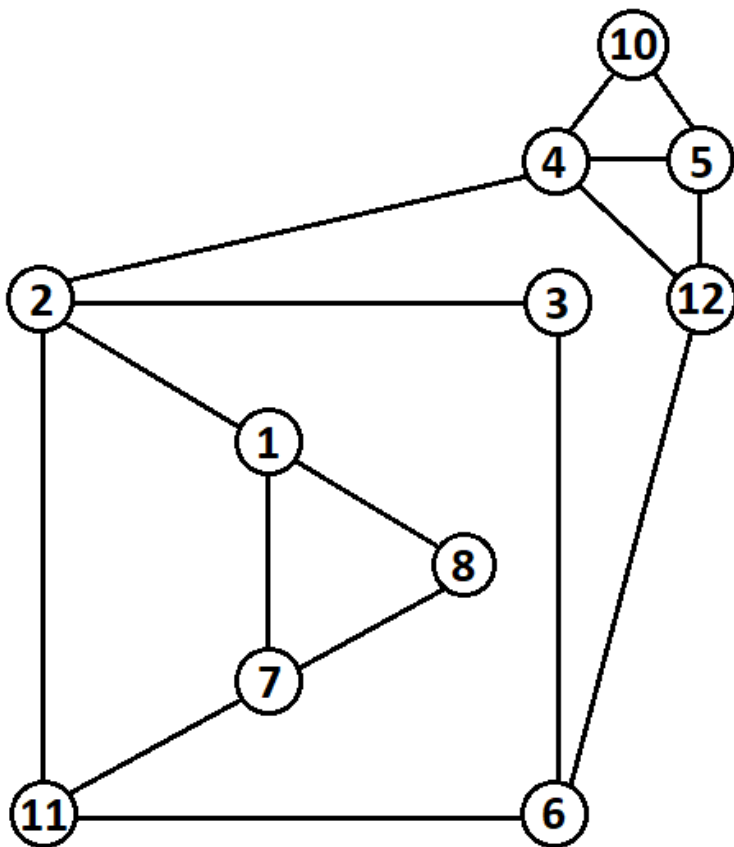
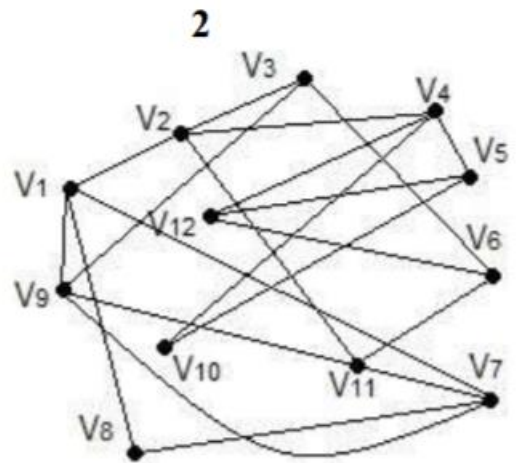
### Завдання №1.

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

2



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





```

        { 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,0,0,3,0,4,0,0 }, /*27*/
        { 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,8,0,0,0,0,4,0,3,0 },
        { 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,3,0,1 }, /*29*/
        { 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,1,0 }
    };

cout << "The adjacency matrix.\n" << endl;
for (int i = 0; i < SIZE; i++)
{
    for (int j = 0; j < SIZE; j++)
    {
        cout << Matrix[i][j] << " ";
    }
    cout << endl;
}

int dis[SIZE]; // відстань
int visited[SIZE]; //відвідані вершини
int minindex, min;
int startpoint = 0;
int split = SIZE/3;
for (int i = 0; i < SIZE; i++)
{
    dis[i] = 10000; //відстань до інших вершин
    visited[i] = 1; //позначити як невідвідано
}
dis[startpoint] = 0;
do {
    minindex = 10000;
    min = 10000;
    for (int i = 0; i < SIZE; i++)
    {
        if ((visited[i] == 1) && (dis[i] < min))
        {
            min = dis[i];
            minindex = i;
        }
    }
    if (minindex != 10000)
    {
        for (int i = 0; i < SIZE; i++)
        {
            if (Matrix[minindex][i] > 0)
            {
                int point = min + Matrix[minindex][i]; //додати знайдену мін вагу
                // до існуючої ваги вершини
                if (point < dis[i]) // порівняти з поточною вагою
                {
                    dis[i] = point;
                }
            }
        }
        visited[minindex] = 0;
    }
} while (minindex < 10000);
// Восстановление пути
int endy;
endy = 30;
int end = endy-1;
int seen[SIZE]; // масив відвіданих вершин
seen[0] = end + 1; // початковий елемент - кінцева вершина
int weight = dis[end]; // вага кінцева вершини
int k = 1; // індекс попередньої
while (split != 0) // пока не початок
{
    for (int i = 0; i < SIZE; i++)
    {

```

```

        if ((Matrix[end][i] != 0)&&(Matrix[end][i] != seen[k])) // якщо вершини
суміжні
        {
            int point = weight - Matrix[end][i];
            if (point == dis[i]) // якщо вага не співпадає
            {
                weight = point;
                end = i; // зберігаємо попередню вершину
                seen[k] = i + 1; // і записуємо її в масив
                k++;
            }
        }
        split--;
    }
}
//Шлях
cout<<"\nThe shortest way from start point to the end\n";
for (int i = k-1; i >= 0; i--)
{
    if (i) cout << seen[i] << " > ";
    else cout << seen[i];
}
cout << "\n Total weight = " << dis[SIZE-1] << endl;
return 0;
}

```

The screenshot shows a terminal window titled "5dlab" containing a 30x30 matrix of integers representing edge weights between vertices. The matrix is mostly zeros, with some non-zero values indicating edges. Below the matrix, the program's output is displayed:

```

The shortest way from start point to the end
1 > 7 > 13 > 19 > 20 > 21 > 22 > 23 > 29 > 30
Total weight = 15
Process finished with exit code 0

```

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