МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота №5

з дисципліни
«Дискретна математика»

Виконав:
студент групи КН-114
Брила Ярослав

Викладач:
Мельникова.Н.І

Львів – 2019 р

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

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

ТЕОРЕТИЧНІ ВІДОМОСТІ ТА ПРИКЛАДИ РОЗВ'ЯЗАННЯ ЗАДАЧ

Задача знаходження найкоротшого шляху з одним джерелом полягає у знаходженні найкоротших(мається на увазі найоптимальніших за вагою) шляхів від деякої вершини(джерела) до всіх вершин графа G. Для розв'язку цієї задачі використовується «жадібний» алгоритм, який називається алгоритмом Дейкстри.

«Жадібними» називаються алгоритми, які на кожному кроці вибирають оптимальний із можливих варіантів.

Задача про найкоротший ланцюг. Алгоритм Дейкстри.

Дано n-вершинний граф $G=(V, E)$, у якому виділено пару вершин $v_0, v^* \in V$, і кожне ребро зважене числом $w(e) \geq 0$. Нехай $X=\{x\}$ – множина усіх простих ланцюгів, що з'єднують v_0 з v^*, $x=(V_x, E_x)$. Цільова функція $F(x)=\sum_{e \in E_x} w(e) \rightarrow \min$. Потрібно знайти найкоротший ланцюг, тобто $x_0 \in X: \; F(x_0)=\min_{x \in X} F(x)$

Перед описом алгоритму Дейкстри подамо визначення термінів "k-а найближча вершина і "дерево найближчих вершин". Перше з цих понять визначається індуктивно так.

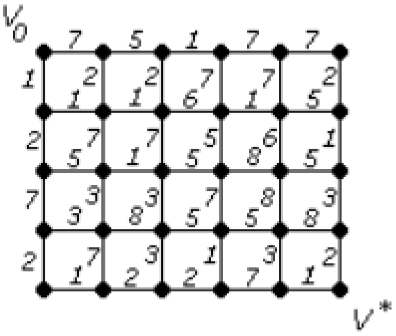
1-й крок індукції. Нехай зафіксовано вершину x_0, E_1 – множина усіх ребер $e \in E$, інцидентних v_0. Серед ребер $e \in E_1$ вибираємо ребро $e(1)=(v_0, v_1)$, що має мінімальну вагу, тобто $w(e(1))=\min_{e \in E_1} w(e)$. Тоді v_1 називаємо першою найближчою вершиною (НВ), число $w(e(1))$ позначаємо $l(1)=l(v_1)$ і називаємо відстанню до цієї НВ. Позначимо $V_1=\{v_0, v_1\}$ – множину найближчих вершин.

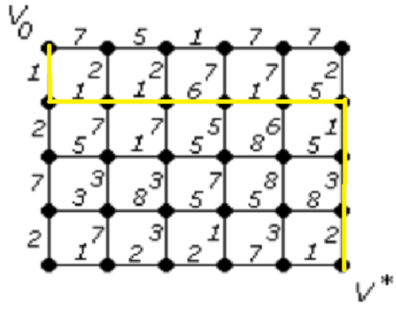
2-й крок індукції. Позначимо E_2 – множину усіх ребер $e=(v',v'')$, $e \in E$, таких що $v' \in V_1$, $v'' \in (V \setminus V_1)$. Найближчим вершинам $v \in V_1$ приписано відстані $l(v)$ до кореня v_0, причому $l(v_0)=0$. Введемо позначення: $\overline{V_1}$ – множина таких вершин $v'' \in (V \setminus V_1)$, що $\exists$ ребра виду $e=(v, v'')$, де $v \in V_1$. Для всіх ребер $e \in E_2$ знаходимо таке ребро $e_2=(v', v_2)$, що величина $l(v')+w(e_2)$ найменша. Тоді v_2 називається другою найближчою вершиною, а ребра e_1, e_2 утворюють зростаюче дерево для виділених найближчих вершин $D_2=\{e_1, e_2\}$.

(s+1)-й крок індукції. Нехай у результаті s кроків виділено множину найближчих вершин $Vs=\{v_0, v_1, \ldots, v_s\}$ і відповідне їй зростаюче дерево $D_s=\{e_1, e_2, \ldots, e_s\}$... Для кожної вершини $v \in V_s$ обчислена відстань $l(v)$ від кореня v_0 до v; $\overline{V_s}$ – множина вершин $v \in (V \setminus V_s)$, для яких існують ребра вигляду $e=(v_r, v)$, де $v_r \in V_s$, $v \in (V \setminus V_s)$. На кроці s+1 для кожної вершини $v_r \in V_s$ обчислюємо відстань до вершини v_r : $L(s+1)(v_r) = l(v_r) + \min\limits_{v^* \in V_s} w(v_r, v^*)$, де min береться по всіх ребрах $e=(v_r, v^*)$, $v^* \in \overline{V}_s$, після чого знаходимо min серед величин $L(s+1)(v_r)$. Нехай цей min досягнуто для вершин v_{r0} і відповідної їй $v^* \in \overline{V}_s$, що назвемо v_{s+1}. Тоді вершину v_{s+1} називаємо (s+1)-ю НВ, одержуємо множину $V_{s+1}=V_s \cup v_{s+1}$ і зростаюче дерево $D_{s+1}=D_s \cup (v_{r0}, v_{s+1})$. (s+1)-й крок завершується перевіркою: чи є чергова НВ v_{s+1} відзначеною вершиною, що повинна бути за умовою задачі зв'язано найкоротшим ланцюгом з вершиною v_0. Якщо так, то довжина шуканого ланцюга дорівнює $l(v_{s+1})=l(v_{r0})+w(v_{r0}, v_{s+1})$; при цьому шуканий ланцюг однозначно відновлюється з ребер зростаючого дерева D_{s+1}. У противному випадку випливає перехід до кроку s+2.

Варіант 2

Завдання № 1. Розв'язати на графах наступні 2 задачі: 1. За допомогою алгоритму Дейкстра знайти найкоротший шлях у графі поміж парою вершин V0 і V* .

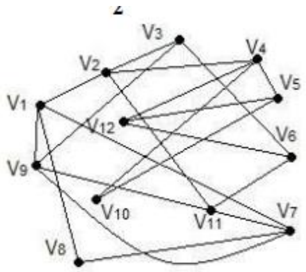


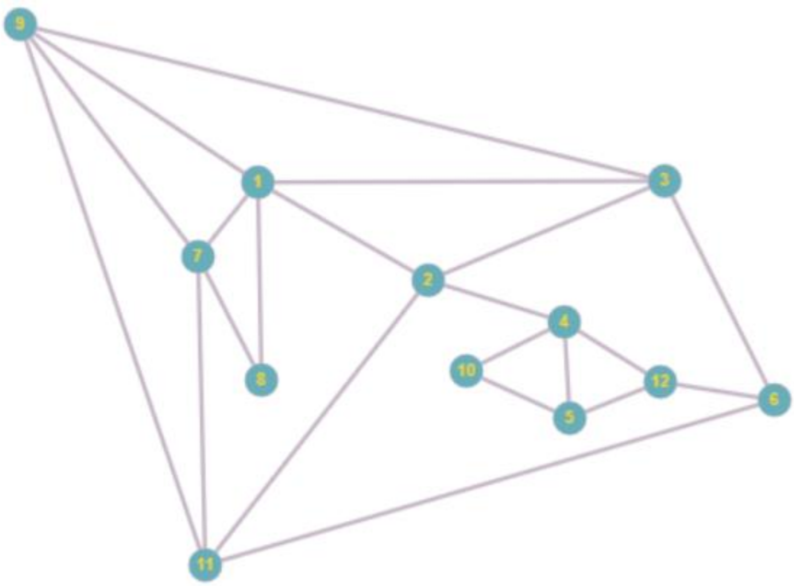


Мінімальна довжина :21

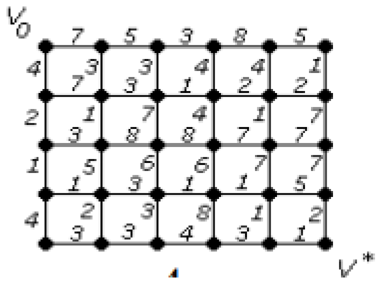
6-7-10-13-16-19-22-23-24-30

Завдання №2.За допомогою γ -алгоритма зробити укладку графа у площині, або довести що вона неможлива.





Завдання №3 Написати програму, яка реалізує алгоритм Дейкстри знаходження найкоротшого шляху між парою вершин у графі. Протестувати розроблену програму на графі згідно свого варіанту.




```cpp
#include <iostream>
#include <vector>
#include <queue>
using namespace std;
priority_queue<pair<int,int>>q;
void back(int x,vector<int> p)
{
    cout<<x+1<<' ';
    if(p[x]!=-1)
        back(p[x],p);
}
int main() {
int n=30,m=49,w,v1,v2;
vector<vector<pair<int,int>>> g(n);
vector<int> p(n),d(n, _Val: 10000);
    d[0]=0;
    p[0]=-1;
    q.push( _Val: { _Val1: 0, _Val2: 0});
    for (int i = 0; i < m; ++i) {
        cin>>v1>>v2>>w;
        v1--;
        v2--;
        g[v1].emplace_back(v2,w);
        g[v2].emplace_back(v1,w);
    }
    while (!q.empty())
    {
        int v=q.top().first, current=-q.top().second;
        q.pop();
        if(current>d[v])continue;
        for (int j = 0; j < g[v].size(); ++j) {
            int to=g[v][j].first, len=g[v][j].second;
            if(d[v]+len<d[to])
            {
                d[to]=d[v]+len;
                p[to]=v;
                q.push({to,-d[to]});
            }
        }
    }
    back( x: 29,p);
}
```

```
7 8 7
8 9 3
9 10 1
10 11 2
11 12 2
13 14 3
14 15 8
15 16 8
16 17 7
17 18 7
19 20 1
20 21 3
21 22 1
22 23 1
23 24 5
25 26 3
26 27 3
27 28 4
28 29 3
29 30 1
1 7 4
7 13 2
13 19 1
19 25 4
2 8 3
8 14 1
14 20 5
20 26 2
3 9 3
9 15 7
15 21 6
21 27 3
4 10 4
10 16 4
16 22 6
22 28 8
5 11 4
11 17 1
17 23 7
23 29 1
6 12 1
12 18 7
18 24 7
24 30 2
1->7->13->19->20->21->22->23->29->30
```

Висновок: Виконуючи цю лабораторну роботу, я закріпив знання алгоритму Дейкстри, а також гамма-вкладки.