

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота № 5

з дисципліни

«Дискретна математика»

Виконала:

студентка групи КН-114

Кемська Юлія

Викладач:

Мельникова Н.І.

Львів – 2019

Варіант -9

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 $V_s=\{v_0, v_1, \dots, v_s\}$ і відповідне їй зростаюче дерево $D_s=\{e_1, e_2, \dots, e_s\}$... 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 : $L(s+1)(v_r)=l(v_r)+\min_{v^* \in \overline{V}_s} w(v_r, v^*)$, де \min

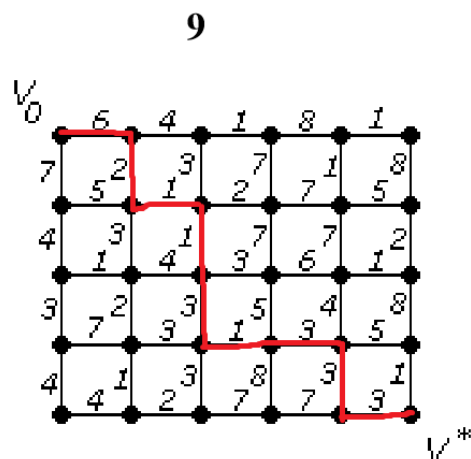
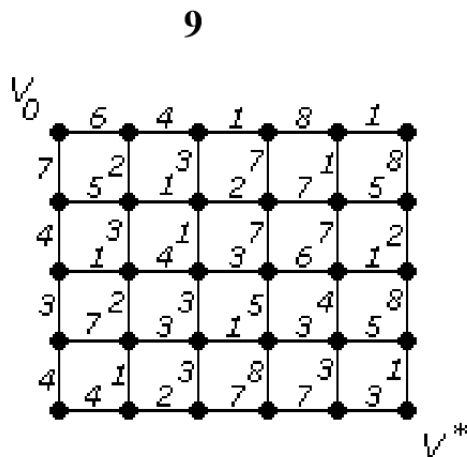
береться по всіх ребрах $e=(v_r, v^*)$, $v^* \in \overline{V}_s$, після чого знаходимо \min серед величин $L(s+1)(v_r)$. Нехай цей \min досягнуто для вершин v_{r^0} і

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

$D_{s+1}=D_s \cup (v_{r^0}, v_{s+1})$. (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 $l(v_{s+1})=l(v_{r^0})+w(v_{r^0}, v_{s+1})$; при цьому шуканий ланцюг однозначно відновлюється з ребер зростаючого дерева D_{s+1} . У протилежному випадку впливає перехід до кроку s+2.

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

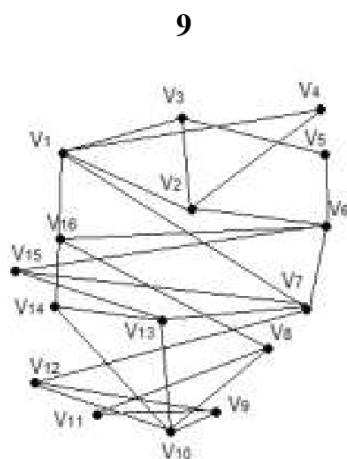
1. За допомогою алгоритму Дейкстри знайти найкоротший шлях у графі поміж парою вершин V_0 і V^* .



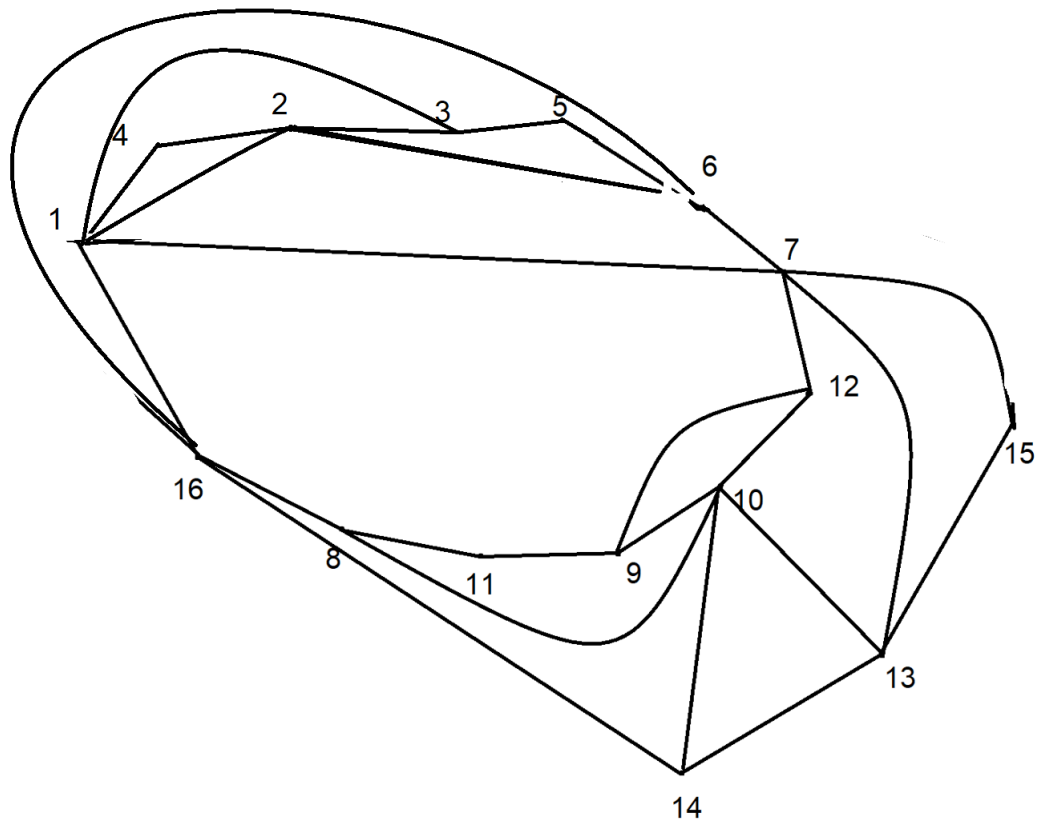
Найменший шлях: 23

$6+2+1+1+3+1+3+3+3$

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



V1 V4 V2 V3 V5 V6 V7 V12 V10 V9 V11 V8 V16



Код реалізації:

```
1  #include <iostream>
2  #include <cstdio>
3  using namespace std;
4  int n,i,j,a[101][101],b[101],c[101]={0},x,y,path[102]={0},from[102]={0};
5  void dejkstra(int v)
6  {
7      c[v]=1;
8      for(i=1;i<=n;i++)
9          if(a[v][i]!=-1 && c[i]==0 && v!=i)
10     {
11         if(b[v]+a[v][i]<b[i]) { path[i]=v;
12         b[i]=min(b[i],b[v]+a[v][i]);from[i] = v;}
13     }
14 }
15 int min_ar()
16 {
17     int min=1000000,nm=0;
18     for(i=1;i<=n;i++)
19         if(min>b[i] && c[i]==0) {min=b[i];nm=i;}
20     return nm;
21 }
22
23 int out(int v){
24     cout<<endl<<"pair"<<endl;
25     cout<<v<<" "<<from[v]<<" ";
26     return from[v];
27 }
28
29 int main()
30 {
31
32     cout<<"Enter number of tops, start and finish tops\n";
33 }
```

```

34     cin>>n>>x>>y;
35     cout<<"Enter matrix\n";
36     for(i=1;i<=n;i++)
37     {
38     for(j=1;j<=n;j++){
39         cin>>a[i][j];
40     }
41
42     b[i]=10000000;
43     }
44     if(a[x][y]==0){cout<<0;return 0;}
45
46     b[x]=0;
47     path[1]=x;
48     int t=1;
49     while(t==1)
50     {
51         t=0;
52         int v=min_ar();
53         if(c[v]==0 && v>0 ) {dejkstra(v); t=1;}
54     }
55
56     cout<<"The shortest way: "<<b[y]<<endl;
57     cout<<"Tops on the shortest way: ";
58     int curr = y;
59     while(curr!=x){
60         curr = out(curr);
61     }
62
63
64     }
65

```

Результат:

```
Enter number of tops, start and finish tops
30 1 30
Enter matrix
-1 6 -1 -1 -1 -1 7 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
6 -1 4 -1 -1 -1 -1 2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 4 -1 1 -1 -1 -1 -1 3 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 1 -1 8 -1 -1 -1 -1 7 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 8 -1 1 -1 -1 -1 -1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 1 -1 -1 -1 -1 -1 -1 8 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
7 -1 -1 -1 -1 -1 -1 5 -1 -1 -1 -1 4 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 2 -1 -1 -1 -1 5 -1 1 -1 -1 -1 -1 3 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 3 -1 -1 -1 -1 1 -1 2 -1 -1 -1 -1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 7 -1 -1 -1 -1 2 -1 7 -1 -1 -1 -1 7 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 1 -1 -1 -1 -1 7 -1 5 -1 -1 -1 -1 7 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 8 -1 -1 -1 -1 5 -1 -1 -1 -1 -1 2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 4 -1 -1 -1 -1 -1 1 -1 -1 -1 -1 3 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 3 -1 -1 -1 -1 1 -1 4 -1 -1 -1 -1 2 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 4 -1 3 -1 -1 -1 -1 3 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 7 -1 -1 -1 -1 3 -1 6 -1 -1 -1 -1 5 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 7 -1 -1 -1 -1 6 -1 1 -1 -1 -1 -1 4 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 2 -1 -1 -1 -1 1 -1 -1 -1 -1 -1 8 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 3 -1 -1 -1 -1 -1 -1 7 -1 -1 -1 4 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 2 -1 -1 -1 -1 7 -1 3 -1 -1 -1 1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 3 -1 -1 -1 -1 3 -1 1 -1 -1 -1 3 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 5 -1 -1 -1 -1 1 -1 3 -1 -1 -1 8 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 4 -1 -1 -1 -1 3 -1 5 -1 -1 -1 3 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 8 -1 -1 -1 -1 5 -1 -1 -1 1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 4 -1 -1 -1 -1 4 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 1 -1 -1 -1 -1 4 -1 2 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 3 -1 -1 -1 -1 2 -1 7 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 8 -1 -1 -1 -1 7 -1 7 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 3 -1 -1 -1 7 -1 3
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 1 -1 -1 -1 3 -1
```

The shortest way: 23

Tops on the shortest way:

pair

30 29

29 23

pair

23 22

pair

22 21

pair

21 15

pair

15 9

pair

9 8

pair

8 2

pair

2 1

Process returned 0 (0x0) execution time : 54.412 s

Press any key to continue.