

Кафедра систем штучного інтелекту

Лабораторна робота №5
з дисципліни
«Дискретна математика»

Виконала:
студентка КН-113
Пеленська Софія
Викладач:
Мельникова Н.І.

Тема:

Знаходження найкоротшого маршруту за алгоритмом Дейкстри. Плоскі планарні граfi.

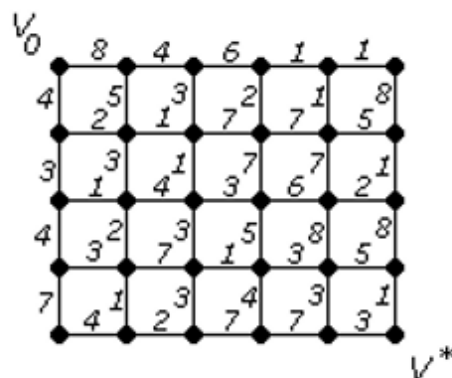
Мета:

Набуття практичних вмінь та навичок з використання алгоритму Дейкстри.

Варіант № 8

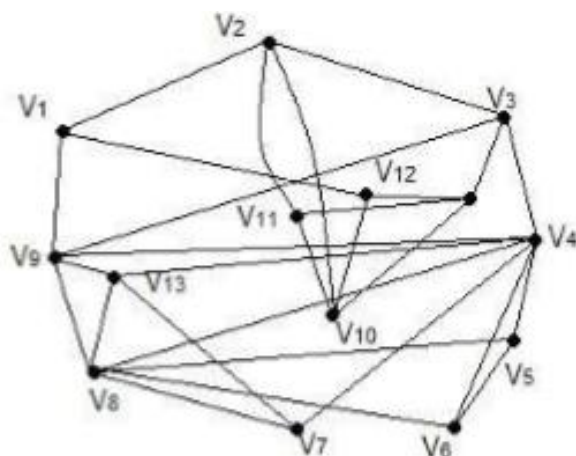
Завдання № 1 : Розв'язати на графах наступні 2 задачі:

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

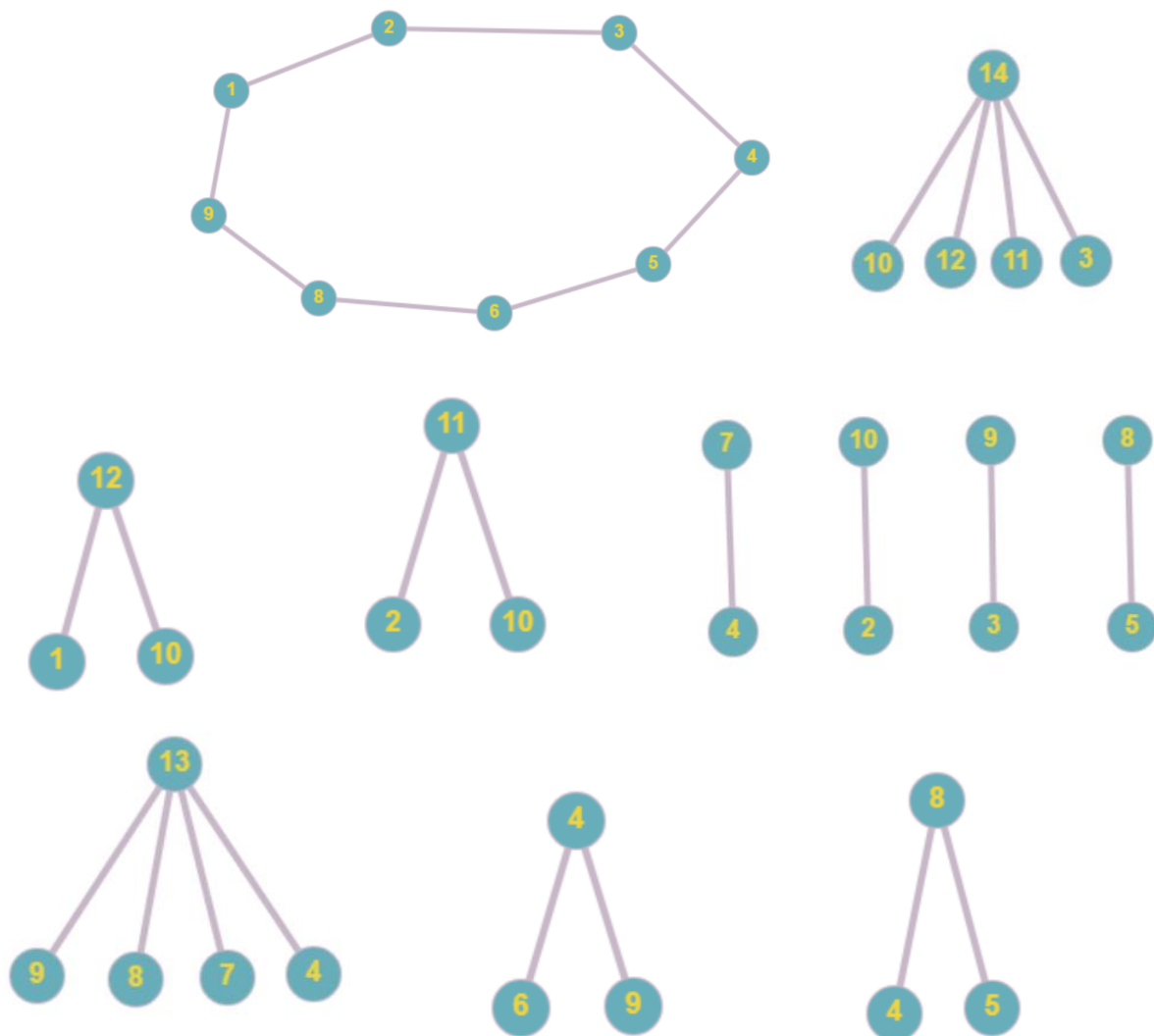


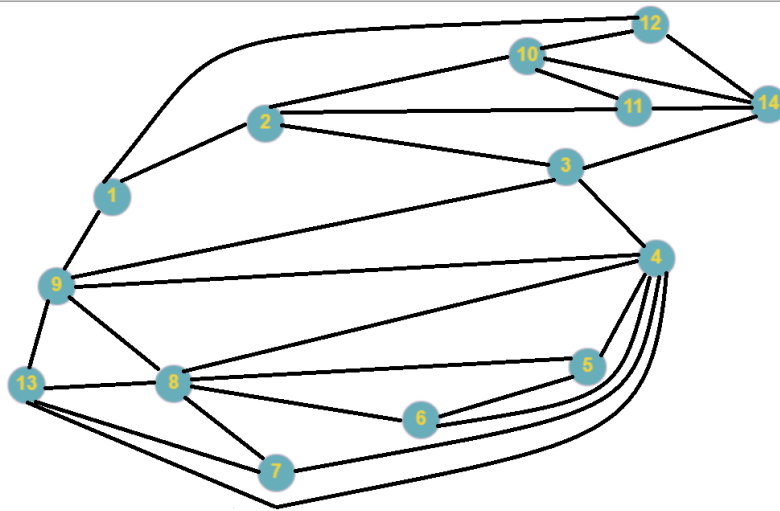
	V0	V1 (8/V0)	V2 (12/V1)	V3 (18/V2)	V4 (19/V3)	V5 (20/V4)
4		8 5 V7 (6/V6)	4 3 V8 (7/V7)	6 2 V9 (20/V3) 14/V8	1 1 V10 (20/V4)	1 8 V11 (28/V8) 25/V10
3		2 3 V13 (9/V7) 8/V12	1 1 V14 (8/V8)	7 7 V15 (21/V9) 11/V14	7 7 V16 (27/V10) 17/V15	5 1 V17 (26/V11) 19/V16
4		1 2 V19 (10/V13)	4 3 V20 (11/V14)	3 5 V21 (16/V15) 12/V20	6 8 V22 (25/V16) 15/V21	2 8 V23 (27/V17) 20/V22
7		3 1 V25 (11/V19)	7 3 V26 (14/V20) 13/V25	1 4 V27 (16/V21)	3 3 V28 (18/V22)	5 1 V* <u>(21/V23)</u> <u>21/V28</u>
	4		2	7	7	3

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

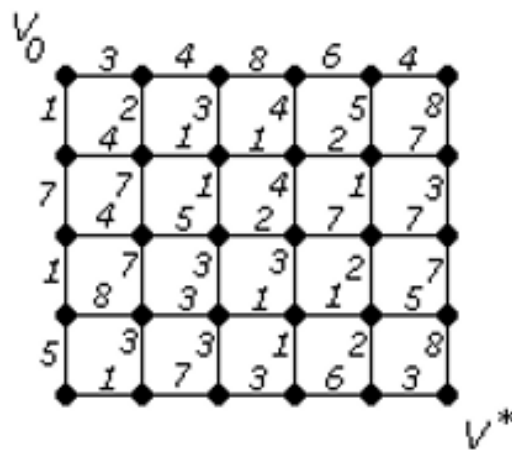


необхідна умова укладки графа — якщо граф не планарний, то він повинен містити більше 4 вершин, степінь яких більше 3, або більше 5 вершин степеня 2. Ця умова не виконується, а отже граф є планарним.





Завдання № 2 : Написати програму, яка реалізує алгоритм Дейкстри знаходження найкоротшого шляху між парою вершин у графі. Протестувати розроблену програму на графі згідно свого варіанту.



Програма :

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int main()
5  {
6      setlocale(LC_ALL, "Ukrainian");
7      int n,i,j;
8      n=30; // кількість вузлів
9      int am[n][n];
10     for (i=0;i<n;i++)
11     {
12         for (j=0;j<n;j++){
13             am[i][j]=0;
14         }
15     }
16
17     am[0][1]=am[1][0]=3;
18     am[0][6]=am[6][0]=1;
19     am[1][2]=am[2][1]=4;
20     am[1][7]=am[7][1]=2;
21     am[2][3]=am[3][2]=8;
22     am[2][8]=am[8][2]=3;
23     am[3][4]=am[4][3]=6;
24     am[3][9]=am[9][3]=4;
25     am[4][5]=am[5][4]=4;
26     am[4][10]=am[10][4]=5;
27     am[5][11]=am[11][5]=8;

```

```

28         am[6][7]=am[7][6]=4;
29         am[6][12]=am[12][6]=7;
30         am[7][8]=am[8][7]=1;
31         am[7][13]=am[13][7]=7;
32         am[8][9]=am[9][8]=1;
33         am[8][14]=am[14][8]=1;
34         am[9][10]=am[10][9]=2;
35         am[9][15]=am[15][9]=4;
36         am[10][11]=am[11][10]=7;
37         am[10][16]=am[16][10]=1;
38         am[11][17]=am[17][11]=3;
39         am[12][13]=am[13][12]=4;
40         am[12][18]=am[18][12]=1;
41         am[13][14]=am[14][13]=5;
42         am[13][19]=am[19][13]=7;
43         am[14][15]=am[15][14]=2;
44         am[14][20]=am[20][14]=3;
45         am[15][16]=am[16][15]=7;
46         am[15][21]=am[21][15]=3;
47         am[16][17]=am[17][16]=7;
48         am[16][22]=am[22][16]=2;
49         am[17][23]=am[23][17]=7;
50         am[18][19]=am[19][18]=8;
51         am[18][24]=am[24][18]=5;
52         am[19][20]=am[20][19]=3;
53         am[19][25]=am[25][19]=3;
54         am[20][21]=am[21][20]=1;

```

```

55         am[20][26]=am[26][20]=3;
56         am[21][22]=am[22][21]=1;
57         am[21][27]=am[27][21]=1;
58         am[22][28]=am[28][22]=2;
59         am[22][23]=am[23][22]=5;
60         am[23][29]=am[29][23]=8;
61         am[24][25]=am[25][24]=1;
62         am[25][26]=am[26][25]=7;
63         am[26][27]=am[27][26]=3;
64         am[27][28]=am[28][27]=6;
65         am[28][29]=am[29][28]=3;
66
67         int numb[n]{-1};
68         int rebra=0;
69         for (i=0;i<n;i++)
70         {
71             for (j=0;j<n;j++)
72             {
73                 if (am[i][j]!=0)
74                     rebra++; // число ребер
75             }
76         }
77         int weight[n]; // массив
78         bool visited[n]; // массив
79         for (i=0;i<n;i++)
80         {
81             weight[i]=10000000;

```

```

82         visited[i]=0;
83     }
84     weight[0]=0;
85     visited[0]=1;
86     int nmin,Vmin1,Vmin2;
87     while (rebra!=0)
88     {
89         nmin=10000000;
90         for (i=0;i<n;i++)
91         {
92             if (visited[i]==1) // уже была посещена
93             {
94                 for (j=0;j<n;j++)
95                 {
96                     if (weight[i]+am[i][j]<nmin && am[i][j]!=0) // на данный момент мин
97                     {
98                         nmin=weight[i]+am[i][j];
99                         Vmin1=i;
100                         Vmin2=j;
101                     }
102                 }
103             }
104         }
105         if (weight[Vmin2]>nmin)
106         {
107             weight[Vmin2]=nmin;
108             numb[Vmin2]=Vmin1;

```

```

109         }
110         visited[Vmin2]=1;
111         am[Vmin1][Vmin2]=am[Vmin2][Vmin1]=0;
112         rebra-=2; //віднімає ребра
113     }
114
115     int endd=29;
116     int way[n];
117     i=0;
118     cout<<"\nВага мінімального шляху = "<<weight[endd]<<endl;
119
120     cout<<"\nШлях:"<<endl;
121     while (endd!=0)
122     {
123         way[i]=endd;
124         endd=numb[endd];
125         i++;
126     }
127     way[i]=0;
128
129     for (i;i>=0;i--)
130     {
131         cout<<way[i];
132         if (i!=0) cout<<"->";
133     }
134     cout<<endl;
135 }

```

Результат програми:

```

Вага мінімального шляху = 17

```

```

Шлях:

```

```

0->1->7->8->9->10->16->22->28->29

```

```

Process returned 0 (0x0)   execution time : 0.068 s
Press any key to continue.

```

Висновок :

На цій лабораторній роботі я набула практичних вмінь та навичок з використання алгоритму Дейкстри.