

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра систем штучного інтелекту

Лабораторна робота №5

З дисципліни

«Дискретна математика»

Виконав :

Студент КН-113

Сайкевич В.А.

Викладач:

Мельникова Н.І.

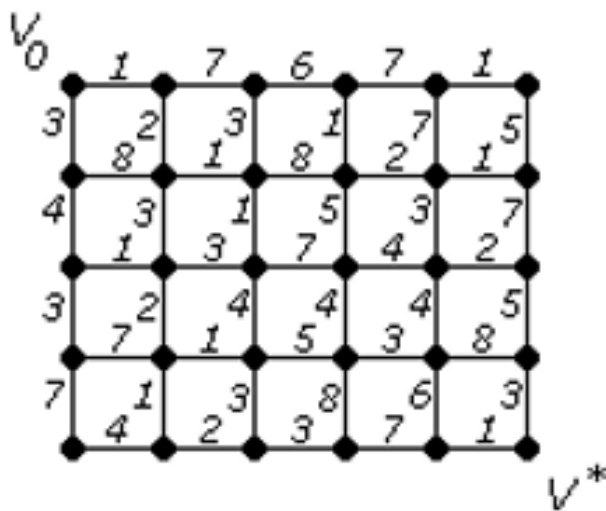
Тема : «Знаходження найкоротшого маршруту за алгоритмом Дейкстри. Плоскі планарні граfi»

Мета : Набуття практичних вмінь та навичок з використання алгоритму Дейкстри.

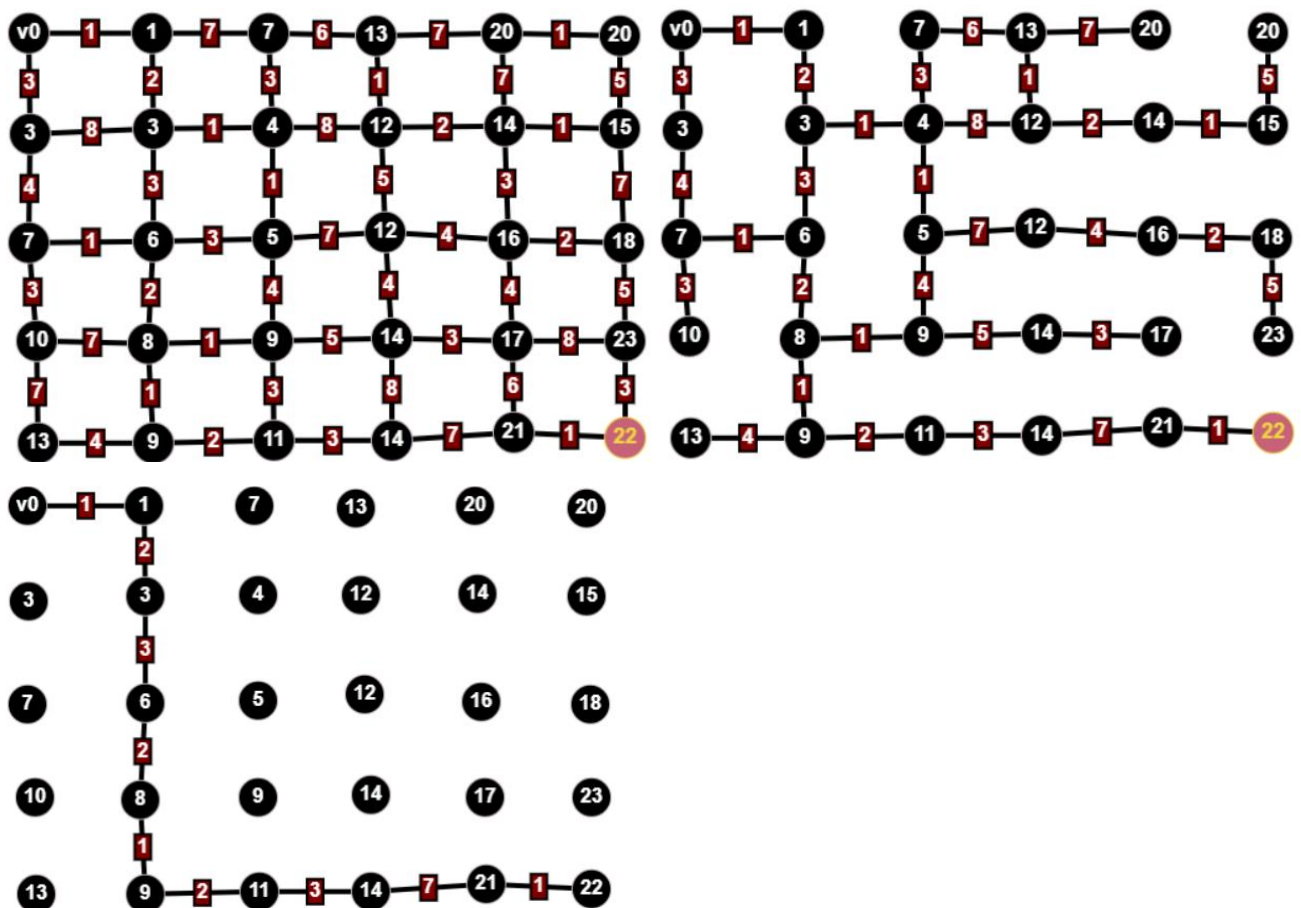
Варіант №14

Завдання 1

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

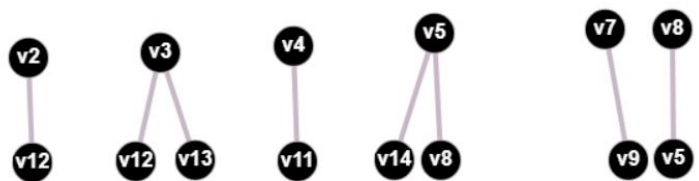
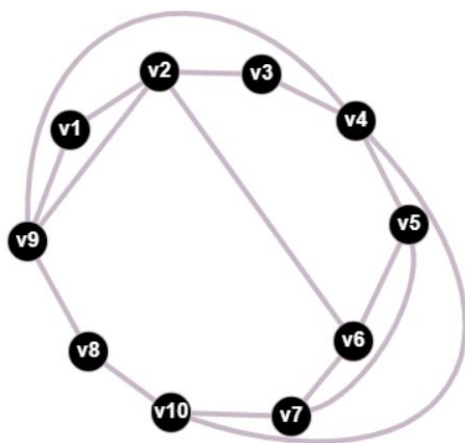
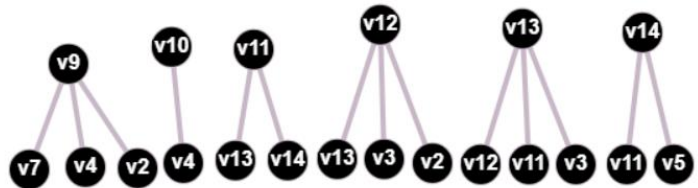
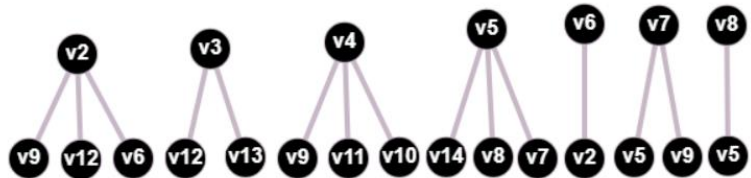
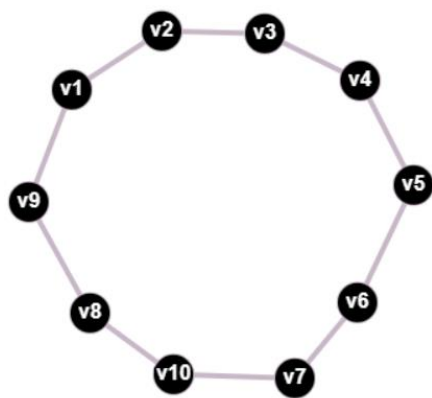
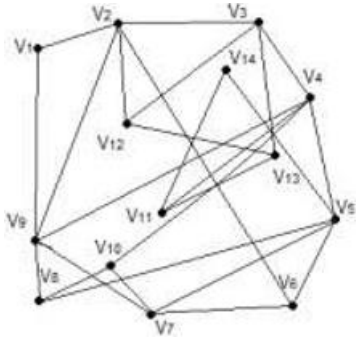


$v_1=1; v_2=3; v_3=3; v_4=4; v_5=5;$
 $v_6=6; v_7=7; v_8=7; v_9=8; v_{10}=9;$
 $v_{11}=9; v_{12}=10; v_{13}=11; v_{14}=12; v_{15}=12;$
 $v_{16}=13; v_{17}=13; v_{18}=14; v_{19}=14; v_{20}=14;$
 $v_{21}=15; v_{22}=16; v_{23}=17; v_{24}=18; v_{25}=20;$
 $v_{26}=20; v_{27}=21; v_{28}=22; v_{29}=23;$



Завдання 2

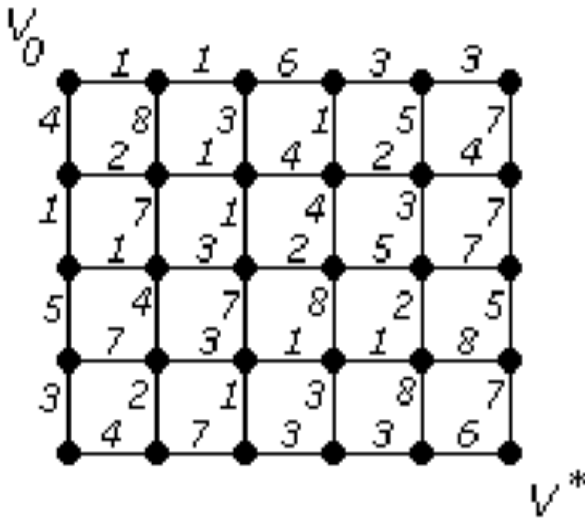
За допомогою γ -алгоритма зробити укладку графа у площині, або довести що вона неможлива:



Враховуючи, що граф містить 6 вершин зі степенем більше 3: v2, v3, v4, v5, v7, v9, його неможливо укласти.

Код

Написати програму, яка реалізує алгоритм Дейкстри знаходження найкоротшого шляху між парою вершин у графі.



```
1  #include <iostream>
2  #include <iomanip>
3
4  using namespace std;
5
6  struct vertex {
7      int x=INT_MAX, left=0, top=0, right=0, bot=0;
8      bool used = 0;
9  };
10
11 void main()
12 {
13     vertex arr[5][6];
14     arr[0][0].x = 0;
15     cout << "Enter lenghts \n";
16     for (int i = 0; i < 4; i++) {
17         for (int j = 0; j < 5; j++) {
18             cout << i + 1 << j + 1 << "-" << i + 1 << j + 2 << " "; cin >> arr[i][j].right; arr[i][j + 1].left = arr[i][j].right;
19             cout << i + 1 << j + 1 << "-" << i + 2 << j + 1 << " "; cin >> arr[i][j].bot; arr[i + 1][j].top = arr[i][j].bot;
20         }
21         cout << i + 1 << "6" << "-" << i + 2 << "6" << " "; cin >> arr[i][5].bot; arr[i + 1][5].top = arr[i][5].bot;
22     }
23     for (int j = 0; j < 5; j++) {
24         cout << "5" << j + 1 << "-" << "5" << j + 2 << " "; cin >> arr[4][j].right; arr[4][j + 1].left = arr[4][j].right;
25     }
26     for (int i = 0; i < 4; i++) {
27         for (int j = 0; j < 5; j++) cout << "#-" << setw(2) << arr[i][j].right << "-";
28         cout << "#\n" << " | | | | | \n";
29         for (int j = 0; j < 6; j++) cout << setw(2) << arr[i][j].bot << " ";
30         cout << "\n" << " | | | | | \n";
31     }
```

```
31 }
32 for (int j = 0; j < 5; j++) cout << "#-" << setw(2) << arr[4][j].right << "-";
33 cout << "#\n\n";
34
35 int min = INT_MAX, mini, minj;
36 for (int n = 0; n < 30; n++, min = INT_MAX) {
37     for (int i = 0; i < 5; i++) for (int j = 0; j < 6; j++) if (arr[i][j].x < min && !arr[i][j].used) { min = arr[i][j].x; mini = i; minj = j; }
38
39     arr[mini][minj].used = 1;
40     if (minj < 5) if (arr[mini][minj + 1].x > arr[mini][minj].x + arr[mini][minj].right) arr[mini][minj + 1].x = arr[mini][minj].x + arr[mini][minj].right;
41     if (minj > 0) if (arr[mini][minj - 1].x > arr[mini][minj].x + arr[mini][minj].left) arr[mini][minj - 1].x = arr[mini][minj].x + arr[mini][minj].left;
42     if (mini < 4) if (arr[mini + 1][minj].x > arr[mini][minj].x + arr[mini][minj].bot) arr[mini + 1][minj].x = arr[mini][minj].x + arr[mini][minj].bot;
43     if (mini > 0) if (arr[mini - 1][minj].x > arr[mini][minj].x + arr[mini][minj].top) arr[mini - 1][minj].x = arr[mini][minj].x + arr[mini][minj].top;
44 }
45 for (int i = 0; i < 4; i++) {
46     for (int j = 0; j < 5; j++) cout << setw(2) << arr[i][j].x << "-" << setw(2) << arr[i][j].right << "-";
47     cout << setw(2) << arr[i][5].x << "\n" << " | | | | | \n";
48     for (int j = 0; j < 6; j++) cout << setw(2) << arr[i][j].bot << " ";
49     cout << "\n" << " | | | | | \n";
50 }
51 for (int j = 0; j < 5; j++) cout << setw(2) << arr[4][j].x << "-" << setw(2) << arr[4][j].right << "-";
52 cout << setw(2) << arr[4][5].x << "\n";
53 }
```

Результати :

Enter lenghts		
11-12 1	31-32 1	51-52 4
11-21 4	31-41 5	52-53 7
12-13 1	32-33 3	53-54 3
12-22 8	32-42 4	54-55 3
13-14 6	33-34 2	55-56 6
13-23 3	33-43 7	#- 1-#- 1-#- 6-#- 3-#- 3-#
14-15 3	34-35 5	
14-24 1	34-44 8	4 8 3 1 5 7
15-16 3	35-36 7	
15-25 5	35-45 2	#- 2-#- 1-#- 4-#- 2-#- 4-#
16-26 7	36-46 5	
21-22 2	41-42 7	1 7 1 4 3 7
21-31 1	41-51 3	
22-23 1	42-43 3	#- 1-#- 3-#- 2-#- 5-#- 7-#
22-32 7	42-52 2	
23-24 4	43-44 1	5 4 7 8 2 5
23-33 1	43-53 1	
24-25 2	44-45 1	#- 7-#- 3-#- 1-#- 1-#- 8-#
24-34 4	44-54 3	
25-26 4	45-46 8	3 2 1 3 8 7
25-35 3	45-55 8	
26-36 7	46-56 7	#- 4-#- 7-#- 3-#- 3-#- 6-#

0- 1- 1- 1- 2- 6- 8- 3-11- 3-14
4 8 3 1 5 7
4- 2- 6- 1- 5- 4- 9- 2-11- 4-15
1 7 1 4 3 7
5- 1- 6- 3- 6- 2- 8- 5-13- 7-20
5 4 7 8 2 5
10- 7-10- 3-13- 1-14- 1-15- 8-23
3 2 1 3 8 7
13- 4-12- 7-14- 3-17- 3-20- 6-26

Висновок:

Ми набули практичних вмінь та навичок з використання алгоритму Дейкстри.