

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА
ПОЛІТЕХНІКА»

Кафедра систем штучного інтелекту



Лабораторна робота №5
З курсу “Дискретна математика”

Виконала:
ст.гр. КН-110
Ямнюк Аліна
Викладач:
Мельникова Н.І.

Лабораторна робота № 5.

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.

Плоскі і планарні графи

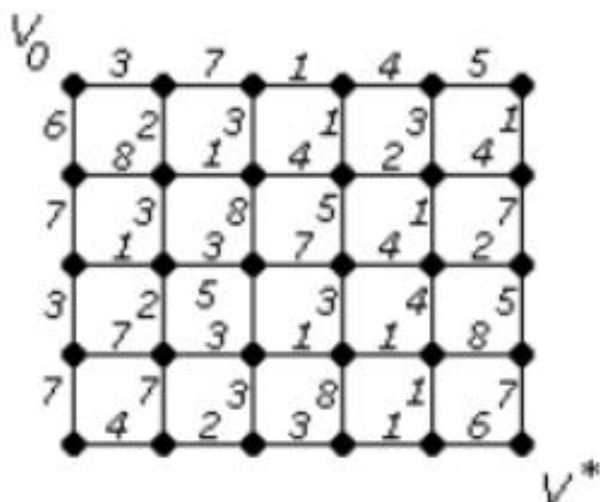
Плоским графом називається граф, вершини якого є точками площини, а ребра – безперервними лініями без самоперетинань, що з'єднують відповідні вершини так, що ніякі два ребра не мають спільних точок крім інцидентної їм обох вершини. Граф називається планарним, якщо він є ізоморфним плоскому графу.

Гранню плоского графа називається максимальна по включенню множина точок площини, кожна пара яких може бути з'єднана жордановою кривою, що не перетинає ребра графа. Границею грані будемо вважати множину вершин і ребер, що належать цій грані.

ІНДИВІДУАЛЬНІ ЗАВДАННЯ

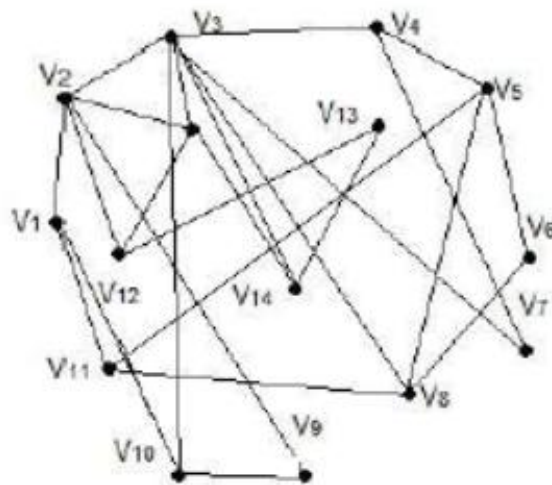
Завдання № 1. Розв'язати на графах наступні 2 задачі:

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

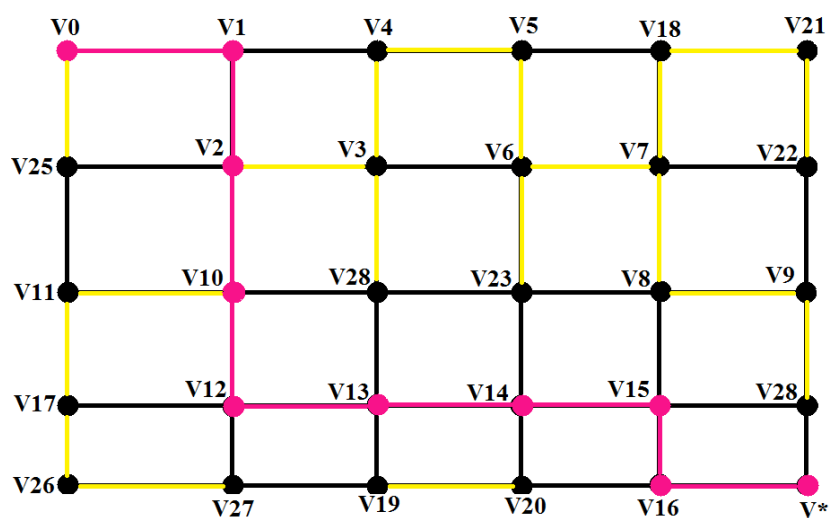


2. За допомогою γ -алгоритма зробити укладку графа у площині, або довести що вона неможлива.

16

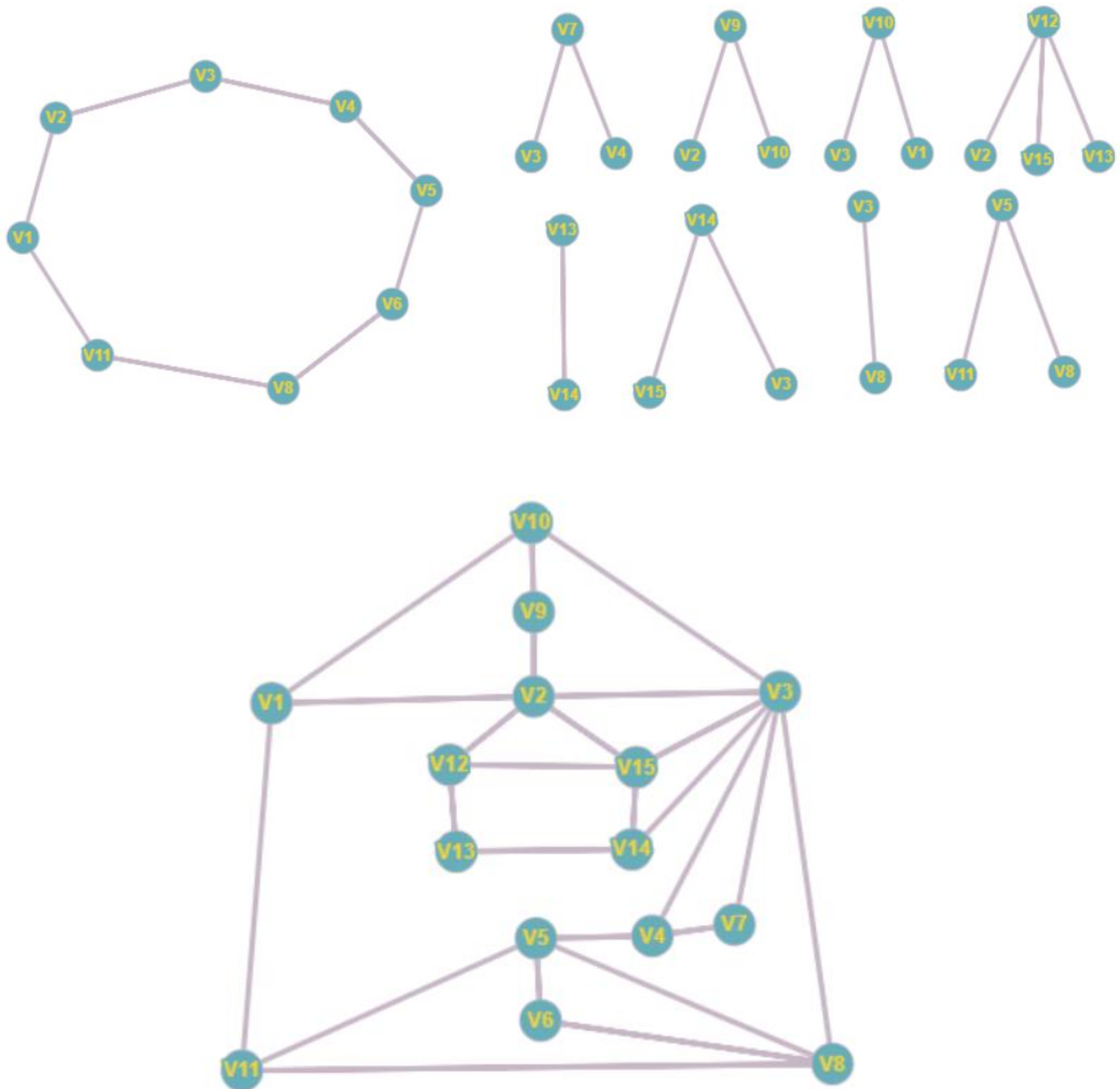


РОЗВ'ЯЗАННЯ



■ - найкоротший шлях

$L(V_0)=0$	$L(V_8)=15$	$L(V_{16})=16$	$L(V_{24})=22$
$L(V_1)=3$	$L(V_9)=17$	$L(V_{17})=12$	$L(V_{25})=6$
$L(V_2)=5$	$L(V_{10})=8$	$L(V_{18})=17$	$L(V_{26})=19$
$L(V_3)=6$	$L(V_{11})=9$	$L(V_{19})=16$	$L(V_{27})=23$
$L(V_4)=10$	$L(V_{12})=10$	$L(V_{20})=19$	$L(V_{28})=14$
$L(V_5)=11$	$L(V_{13})=13$	$L(V_{21})=22$	$L(V^*)=22$
$L(V_6)=12$	$L(V_{14})=14$	$L(V_{22})=23$	
$L(V_7)=14$	$L(V_{15})=15$	$L(V_{23})=17$	



Код програми:

```

1. #include<stdio.h>
2.
3. #define INFINITY 9999
4. #define MAX 30
5.
6. void dejkstra(int G[MAX][MAX], int n, int startnode);
7.
8. int main()
9. {
10.     int G[MAX][MAX], i, j, n, u;
11.     printf("Enter number of vertices:");
12.     scanf("%d", &n);
13.     for (i = 0; i < n; i++)
14.         for (j = 0; j < n; j++)
15.             G[i][j] = 0;

```

```

16.
17. printf("Enter graph (If it is the end, enter number 111):\n");
18. int st, nd, weight;
19. while(1)
20. {
21.     printf("1st vertex: ");
22.     scanf("%d", &st);
23.     while (((st < 0) || (st > n - 1)) && (st != 111))
24.     {
25.         printf("Retry: ");
26.         scanf ("%d", &st);
27.     }
28.     if (st == 111)
29.         break;
30.     printf("2nd vertex: ");
31.     scanf("%d", &nd);
32.     while (((nd < 0) || (nd > n - 1)) && (nd != 111))
33.     {
34.         printf("Retry: ");
35.         scanf("%d", &nd);
36.     }
37.     if (nd == 111)
38.         break;
39.     printf("Weight: ");
40.     scanf("%d", &weight);
41.     while (weight <= 0)
42.     {
43.         printf("Retry: ");
44.         scanf("%d", &weight);
45.     }
46.     if (weight == 111)
47.         break;
48.     if ((st != 111) && (nd != 111) && (weight != 111))
49.     {
50.         G[st][nd] = weight;
51.         G[nd][st] = weight;
52.     }
53. }
54. printf("\nEnter the starting node:");
55. scanf("%d", &u);
56. dejkstra(G, n, u);
57. printf("\n");
58. return 0;
59. }
60.
61. void dejkstra(int G[MAX][MAX], int n, int startnode)
62. {
63.
64.     int cost[MAX][MAX], distance[MAX], pred[MAX];
65.     int visited[MAX], count, mindistance, nextnode, i, j;
66.
67.     for (i = 0; i < n; i++)
68.         for (j = 0; j < n; j++)
69.             if (G[i][j] == 0)
70.                 cost[i][j] = INFINITY;

```

```

71.         else
72.             cost[i][j] = G[i][j];
73.
74.     for(i = 0; i < n; i++)
75.     {
76.         distance[i] = cost[startnode][i];
77.         pred[i] = startnode;
78.         visited[i] = 0;
79.     }
80.
81.     distance[startnode] = 0;
82.     visited[startnode] = 1;
83.     count = 1;
84.
85.     while (count < n - 1)
86.     {
87.         mindistance = INFINITY;
88.
89.         for (i = 0; i < n; i++)
90.             if (distance[i] < mindistance && !visited[i])
91.             {
92.                 mindistance = distance[i];
93.                 nextnode = i;
94.             }
95.
96.         visited[nextnode] = 1;
97.         for (i = 0; i < n; i++)
98.             if (!visited[i])
99.                 if (mindistance + cost[nextnode][i] < distance[i])
100.                {
101.                    distance[i] = mindistance + cost[nextnode][i];
102.                    pred[i] = nextnode;
103.                }
104.         count++;
105.     }
106.
107.     for (i = 29; i < n; i++)
108.         if(i != startnode)
109.         {
110.             printf("\nDistance from 0 to %d = %d", i, distance[i]);
111.             printf("\nThe shortest way = %d", i);
112.
113.             j = i;
114.             do
115.             {
116.                 j = pred[j];
117.                 printf("<-%d",j);
118.             } while (j != startnode);
119.         }
120.     }

```

Результат роботи програми:

```
Enter number of vertices:30
Enter graph:
1st vertex: 23
2nd vertex: 29
Weight: 7

Starting node is V0
Distance from V0 to V* = 22
The shortest way = 29<-28<-22<-21<-20<-19<-13<-7<-1<-0
```

Висновок: набула практичних вмінь та навичок з використання алгоритму Дейкстри.