

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА**  
**ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

**Розрахункова робота**

з дисципліни

«Дискретна математика»

**Виконала:**

Студентка

групи КН-112

Гудз Юлія

**Викладач:**

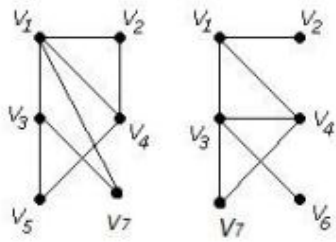
Мельникова Н.І.

Львів – 2019 р

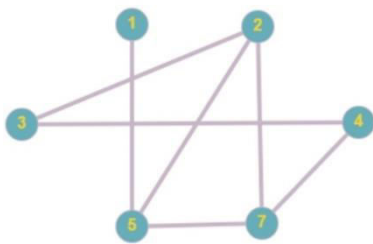
## Завдання № 1

Виконати наступні операції над графами: 1) знайти доповнення до першого графу, 2) об'єднання графів, 3) кільцеву сумму  $G1$  та  $G2$  ( $G1+G2$ ), 4) розмножити вершину у другому графі, 5) виділити підграф  $A$  - що складається з 3-х вершин в  $G1$  6) добуток графів

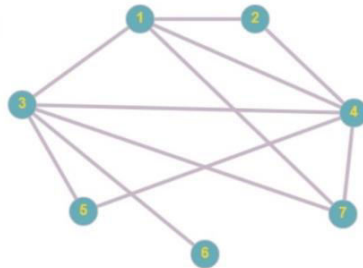
9)



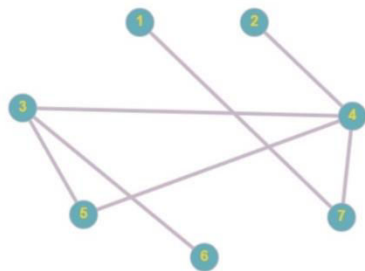
1)



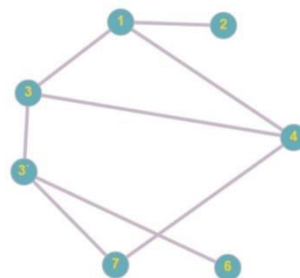
2)



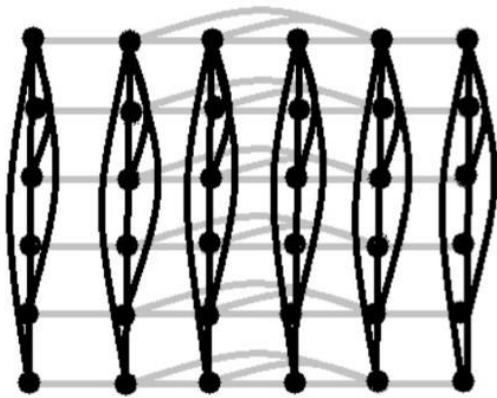
3)



4)



6)

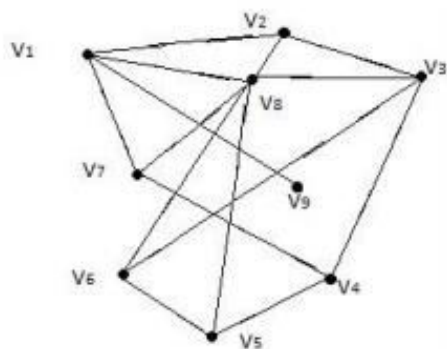


7)

## Завдання № 2

Скласти таблицю суміжності для орграфа

9)



	V1	V2	V3	V4	V5	V6	V7	V8	V9
V1	0	1	0	0	0	0	1	1	1
V2	1	0	1	0	0	0	0	1	0
V3	0	1	0	1	0	1	0	1	0
V4	0	0	1	0	1	0	1	0	0
V5	0	0	0	1	0	1	0	1	0
V6	0	0	1	0	1	0	0	1	0
V7	1	0	0	1	0	0	0	1	0
V8	1	1	1	0	1	1	1	0	0
V9	1	0	0	0	0	0	0	0	0

## Завдання № 3

Для графа з другого завдання знайти діаметр

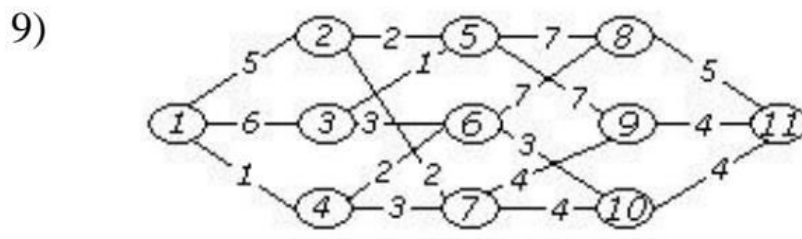
$d=3$

### Завдання № 3

Для графа з другого завдання виконати обхід дерева вглиб (варіант закінчується на непарне число)

### Завдання № 5

Знайти двома методами (Краскала і Прима) мінімальне остове дерево графа.



Алгоритм Прима:

```
main.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      setlocale(LC_ALL, "Ukrainian");
7      int v, N = 0;
8      int MIN = 0, r, l;
9      v=11;
10     int matrix[v][v]
11     {{0,5,6,1,0,0,0,0,0,0,0},
12      {5,0,0,0,2,0,2,0,0,0,0},
13      {6,0,0,0,1,3,0,0,0,0,0},
14      {1,0,0,0,0,2,3,0,0,0,0},
15      {0,2,1,0,0,0,0,7,0,0,0},
16      {0,0,3,2,0,0,0,7,0,3,0},
17      {0,2,0,3,0,0,0,0,4,4,0},
18      {0,0,0,0,7,0,0,0,0,0,5},
19      {0,0,0,0,7,0,4,0,0,0,4},
20      {0,0,0,0,0,3,4,0,0,0,4},
21      {0,0,0,0,0,0,0,5,4,4,0}};
22
23     int reb[v-1][2];
24     int verish[v];
25     bool check =false;
26
27     cout<<"Матриця:"<<endl;
28     for (int i = 0; i < v; i++) {
29         cout<<endl;
30     }
31 }
```

```

28 for (int i = 0; i < v; i++) {
29     cout<<endl;
30     for (int j = 0; j < v; j++) {
31         cout<<matrix[i][j]<<" ";
32     }
33 }
34 cout<<endl;
35
36 versh[N] = 1;
37 N++;
38
39 for (int i = 0; N < v; i++) {
40     for (int j = 0; j < N; j++) {
41         for (int a = 0; a < v; a++) {
42             for (int m = 0; m < N; m++) {
43                 if (versh[m] == a + 1) {
44                     check = true;
45                 }
46             }
47
48             if (check)
49                 { check = false;
50                   continue; }
51             if (MIN == 0 && matrix[versh[j] - 1][a] > 0) {
52                 MIN = matrix[versh[j] - 1][a];
53                 r = reb[N - 1][0] = versh[j];
54                 l = reb[N - 1][1] = a + 1;
55                 continue;
56             }

```

```
mscpp
58     if (matrix[versh[j] - 1][a] > 0 && matrix[versh[j] - 1][a] < MIN) {
59         MIN = matrix[versh[j] - 1][a];
60         r=reb[N-1][0] = versh[j];
61         l = reb[N-1][1] = a + 1;
62     }
63 }
64 }
65 versh[N] = 1;
66 N++;
67 MIN = 0;
68 }
69 cout<<endl<< "V: { ";
70
71 for (int j = 0; j < v; j++) {
72     cout << versh[j]<<" ";
73 }
74
75 cout << " }";
76 cout <<endl<< "E:{ ";
77
78 for (int j = 0; j < v - 1; j++) {
79     cout << "(" << reb[j][0]<<" "<< reb[j][1]<<" )";
80 }
81 cout << " }";
82 cout<<endl<<"Ваги ребер:"<<endl;
83 int sum=0;
84 for (int j=0;j<v-1; j++){
85     int k,m;
86     k=reb[j][0];
87     m=reb[j][1];
88     cout<<matrix[k-1][m-1]<<" ";
89     sum+=matrix[k-1][m-1];
90 }
91 cout<<endl<<endl<<"Мінімальна вага остового дерева: "<<sum<<endl<<endl;
92
93 return 0;
94 }
95
```

Активация Windows

```
C:\Users\user\Labs\laba.4.dm\bin\Debug\laba.4.exe
Матриця:
0 5 6 1 0 0 0 0 0 0
5 0 0 0 2 0 2 0 0 0
6 0 0 0 1 3 0 0 0 0
1 0 0 0 0 2 3 0 0 0
0 2 1 0 0 0 0 7 7 0
0 0 3 2 0 0 0 7 0 3
0 2 0 3 0 0 0 0 4 4
0 0 0 0 7 7 0 0 0 5
0 0 0 0 7 0 4 0 0 4
0 0 0 0 3 4 0 0 0 4
0 0 0 0 0 0 5 4 4 0

V: { 1, 4, 6, 7, 2, 5, 3, 10, 9, 11, 8, }
E:{ (1,4),(4,6),(4,7),(7,2),(2,5),(5,3),(6,10),(7,9),(10,11),(11,8),}

Ваги ребер:
1, 2, 3, 2, 2, 1, 3, 4, 4, 5,

Мінімальна вага остового дерева: 27

Process returned 0 (0x0)   execution time : 0.304 s
Press any key to continue.
```

Алгоритм Краскала:

```
main.cpp [laba.d.m.sofiya] - Code::Blocks 17.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Management
Workspace
laba.4.d.m
Sources
laba.d.m.sofiya
Sources
main.cpp
main.cpp
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4
5 void dubl(int n, int A[11][11]) {
6     for (int i = 0; i < 11; i++) {
7         for (int j = 0; j < 11; j++) {
8             if (i < j) {
9                 A[i][j] = 0;
10            }
11        }
12    }
13
14    int create(int n, int A[11][11]) {
15        for (int i = 0; i < 11; i++) {
16            for (int j = 0; j < 11; j++) {
17                A[i][j] = 0;
18            }
19        }
20        for (int i = 0; i < 11; i++) {
21            A[i][i] = i+1;
22        }
23        return A[11][11];
24    }
25
26    int no(int n, int A[11][11], int t, int z) {
27        int tm, tm2;
28        for (int i = 0; i < 11; i++) {
```

```
28         for (int i = 0; i < 11; i++) {
29             tm = tm2 = 0;
30             for (int j = 0; j < 11; j++) {
31                 if (A[i][j] == t) {
32                     tm = j;
33                 }
34             }
35             for (int f = 0; f < 11; f++) {
36                 if (A[i][f] == z) {
37                     tm2 = f;
38                 }
39             }
40             if (tm <= tm2) {
41                 return 0;
42             }
43         }
44         return 1;
45     }
46
47     void add(int n, int A[11][11], int t, int z) {
48         int scn;
49         for (int i = 0; i < 11; i++) {
50             for (int j = 0; j < 11; j++) {
51                 if (A[i][j] == z) {
52                     scn = j;
53                 }
54             }
55         }
```

```
56         for (int i = 0; i < 11; i++) {
57             for (int j = 0; j < 11; j++) {
58                 if (A[i][j] == t) {
59                     for (int k = 0; k < 11; k++) {
60                         A[i][k] = A[scn][k];
61                         A[scn][k] = 0;
62                     }
63                 }
64             }
65         }
66     }
67
68     int main() {
69         int graph[11][11] {
70             {0,1,2,3,0,0,0,0,0,0,0},
71             {1,0,0,0,4,0,2,0,0,0,0},
72             {0,0,0,0,7,5,0,0,0,0,0},
73             {3,0,0,0,0,2,3,0,0,0,0},
74             {0,4,7,0,0,0,0,4,6,0,0},
75             {0,0,5,2,0,0,0,7,0,3,0},
76             {0,2,0,3,0,0,0,0,0,5,0},
77             {0,0,0,0,4,7,0,0,0,0,4},
78             {0,0,0,0,6,0,8,0,0,0,1},
79             {0,0,0,0,3,7,0,0,0,0,4},
80             {0,0,0,0,0,0,0,8,1,8,0}
81         };
82
83         for (int j = 0; j < 11; j++) {
```

The screenshot shows the Code::Blocks IDE with a C++ project named 'laba.dm.sofiya'. The main.cpp file contains a program that initializes a graph, creates a tree, and prints the results. The output window shows the following data:

```
0 1 2 3 0 0 0 0 0 0
1 0 0 4 0 2 0 0 0 0
2 0 0 7 5 0 0 0 0 0
3 0 0 0 2 3 0 0 0 0
0 4 7 0 0 0 4 6 0 0
0 0 5 2 0 0 0 7 0 3
0 2 0 3 0 0 0 5 7 0
0 0 0 4 7 0 0 0 0 4
0 0 0 6 0 5 0 0 0 1
0 0 0 0 3 7 0 0 0 4
0 0 0 0 0 4 1 4 0

New Tree: 1,2 9,11 1,3 2,7 4,6 1,4 4,7 6,10 2,5 5,8 8,11 10,11 3,6 7,9 5,9 3,5 6,8 7,10
Process returned 0 (0x0)   execution time : 0.097 s
Press any key to continue.
```

## Завдання № 7

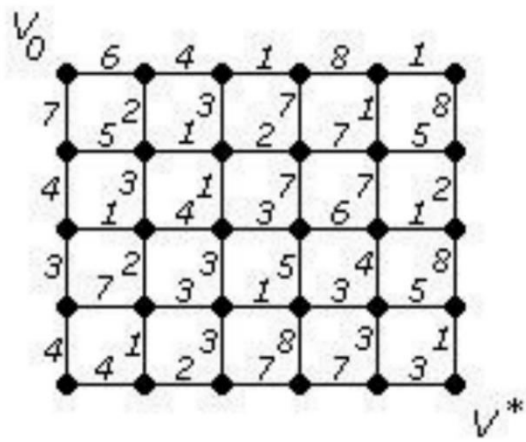
За допомогою алгоритму Дейкстри знайти найкоротший



шлях у графі між парою вершин  $V_0$  і  $V$ .

9)

1



```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      setlocale(LC_ALL, "Ukrainian");
6      int v;
7      cout << "Кількість вершин графа : ";
8      cin >> v;
9      int graph[v][v];
10     for (int i = 0; i < v; i++) {
11         for (int j = 0; j < v; j++) {
12             graph[i][j] = 0;
13         }
14     }
15     int a, b, c;
16     cout << "Заповнити таблицю (щоб завершити вводити 0)" << endl;
17     cout << "вершина 1 | вершина 2 | data : " << endl;
18     while (1) {
19         cin >> a;
20         if (a == 0) {
21             break;
22         }
23         cin >> b >> c;
24         graph[a - 1][b - 1] = graph[b - 1][a - 1] = c;
25     }
26
27     int verts[v][2];
28     int steps[v];
29
30     int first = 1;
31     for (int i = 0; i < v; i++) {
32         if (i == first - 1) {
33             verts[i][0] = 0;
34             verts[i][1] = 1;
35         }
36         else {
37             verts[i][0] = 999;
38             verts[i][1] = 1;
39         }
40     }
41     steps[first - 1] = 0;
42
43     int m;
44     for (int i = 0; i < v; i++) {
45         for (int i = 0; i < v; i++) {
46             if (verts[i][1]) {
47                 m = i;
48                 break;
49             }
50         }
51     }
52     for (int i = 1; i < v; i++) {
53         if (verts[m][0] >= verts[i][0] && verts[i][1] == 1) {
54             m = i;
55         }
56     }
57     for (int j = 0; j < v; j++) {
58         if (graph[m][j]) {
59             if (verts[j][0] > verts[m][0] + (graph[m][j])) {
60                 verts[j][0] = verts[m][0] + (graph[m][j]);
61                 steps[j] = m;
62             }
63         }
64     }
65     verts[m][1] = 0;
66
67     int last = 30;
68     cout << "Мінімальний шлях: ";
69     cout << verts[last - 1][0];
70     cout << endl << last << " ";
71     last--;
72     while (steps[last] != first - 1) {
73         cout << steps[last] + 1 << " ";
74         last = steps[last];
75     }
```

```
C:\Users\user\Labs\laba.dm.5\bin\Debug\laba.dm.exe
Кількість вершин графа : 30
Заповніть таблицю(щоб завершити введіть 0)
|вершина 1|вершина 2|вага| :
1 2 6
2 3 4
3 4 1
4 5 8
5 6 1
7 8 5
8 9 1
9 10 2
10 11 7
11 12 5
13 14 1
14 15 4
15 16 3
16 17 6
17 18 1
19 20 7
20 21 3
21 22 1
22 23 3
23 24 5
25 26 4
26 27 2
27 28 7
28 29 7
29 30 3
1 7 7
2 8 2

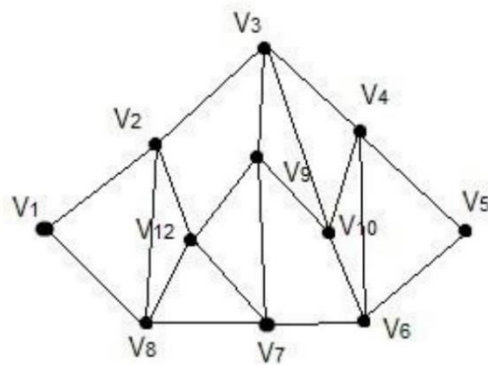
C:\Users\user\Labs\laba.dm.5\bin\Debug\laba.dm.exe
2 8 2
3 9 3
4 10 7
5 11 1
6 12 8
7 13 4
8 14 3
9 15 1
10 16 7
11 17 7
12 18 2
13 19 3
14 20 2
15 21 3
16 22 5
17 23 4
18 24 8
19 25 4
20 26 1
21 27 3
22 28 8
23 29 3
24 30 1
0
Мінімальний шлях: 23
30; 29; 23; 22; 21; 15; 9; 8; 2; 1

Process returned 0 (0x0)   execution time : 46.013 s
Press any key to continue.
```

## Завдання № 8

Знайти ейлеровий цикл в ейлеровому графі двома методами: а) Флері; б) елементарних циклів.

9)



## А) Флері

```

1  #include<iostream>
2  #include<vector>
3  using namespace std;
4
5  void Find(int v, vector< vector<int> >> G, int N)
6  {
7      int i;
8      for (i = 0; i < N; i++)
9          if ((*G)[v][i])
10             {
11                 (*G)[v][i] = (*G)[i][v] = 0;
12                 Find(i, G, N);
13             }
14      cout << v+1 << " ";
15  }
16
17  int main()
18  {
19      setlocale(LC_ALL, "ukr");
20      int N = 0;
21      cout << "Введіть кількість вершин:"; cin >> N;
22      vector< vector<int> > G(N, vector<int>(N));
23      for (int i = 0; i < N; ++i)
24          for (int j = 0; j < N; ++j)
25              cin >> G[i][j];
26
27      int pow, p, q, sum;
28      pow = 1;
29      for (p = 0; p < N; p++)
30      {
31          sum = 0;
32          for (q = 0; q < N; q++)
33              sum += G[p][q];
34          if (sum % 2)
35              pow = 0;
36      }
37      cout << endl;
38      if (pow)
39          Find(0, G, N);
40      else
41          cout << "Неможливо виконати алгоритм Флері" << endl;
42      cout << endl;
43      return 0;
44  }

```

```
C:\Users\user\Labs\fleri\bin\Debug\fleri.exe
Введіть кількість вершин:11
0 1 0 0 0 0 0 1 0 0 0
1 0 1 0 0 0 0 1 0 0 1
0 1 0 1 0 0 0 0 1 1 0
0 0 1 0 1 1 0 0 0 1 0
0 0 0 1 0 1 0 0 0 0 0
0 0 0 1 1 0 1 0 0 1 0
0 0 0 0 0 1 0 1 1 0 1
1 1 0 0 0 0 1 0 0 0 1
0 0 1 0 0 0 1 0 0 1 1
0 0 1 1 0 1 0 0 1 0 0
0 1 0 0 0 0 1 1 1 0 0

1 8 11 7 8 2 11 9 10 6 7 9 3 10 4 6 5 4 3 2 1

Process returned 0 (0x0)   execution time : 2.131 s
Press any key to continue.
```

## Завдання №9

Спростити формули (привести їх до скороченої ДНФ).

$$9. (x \rightarrow y) \cdot (y \rightarrow z) \rightarrow (x \rightarrow z).$$

$$(X \rightarrow Y)(Y \rightarrow Z) \rightarrow (X \rightarrow Z)$$

$$(\neg X \vee Y)(Y \rightarrow Z) \rightarrow (X \rightarrow Z)$$

$$(X \vee Y)(\neg Y \vee Z) \rightarrow (X \rightarrow Z)$$

$$(\neg X \neg Y \vee YZ) \rightarrow (X \rightarrow Z)$$

$$(\neg X \neg Y \vee YZ) \rightarrow (\neg X \vee Z)$$