МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних наук та інформаційних технологій
Кафедра систем штучного інтелекту

Розрахункова робота
з курсу “Дискретна математика ”

Виконав:
студент групи КН-112
Хедик Адольф

Викладач:

Мельникова Н.І.

Варіант 12

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

1) знайти доповнення до першого графу,

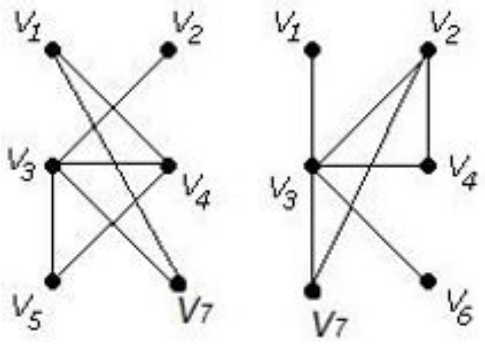
2) об’єднання графів,

3) кільцеву суму G1 та G2 (G1+G2),

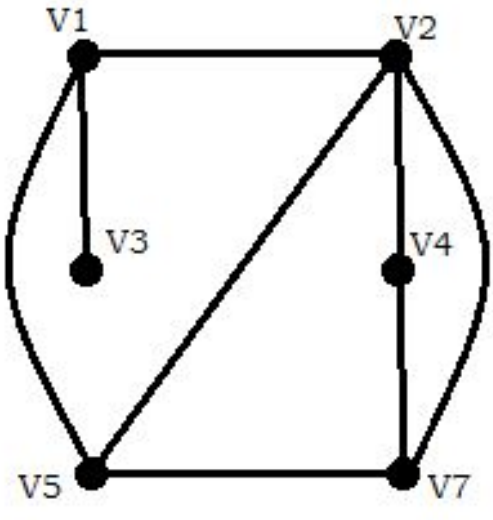
4) розщепити вершину у другому графі,

5) виділити підграф А, що складається з 3-х вершин в G1 і знайти стягнення А в G1 (G1\ A),

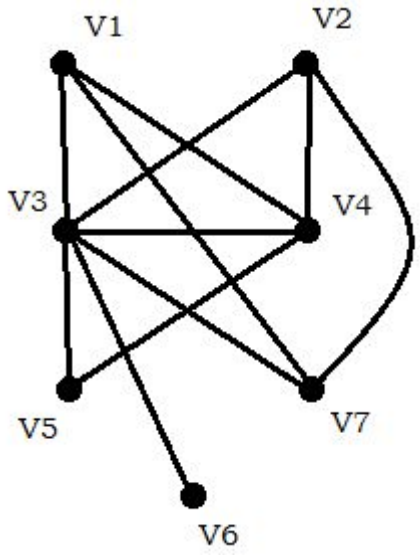
6) добуток графів



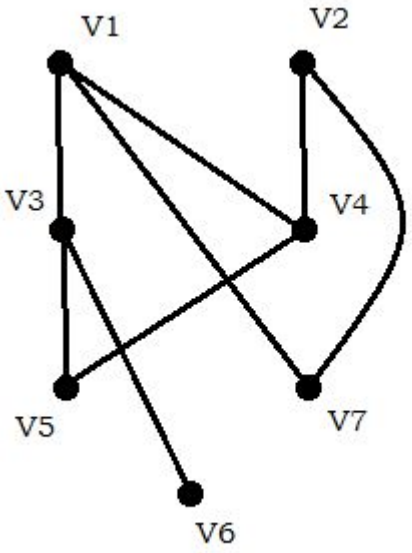
1) доповнення до першого графа

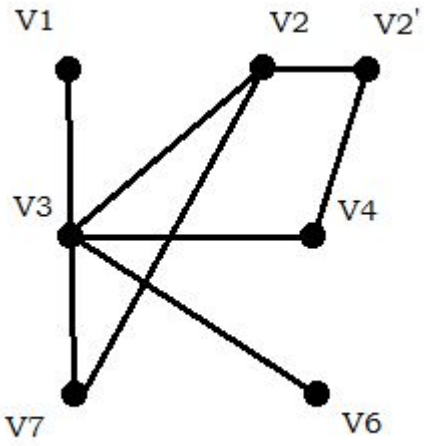


2) об’єднання графів

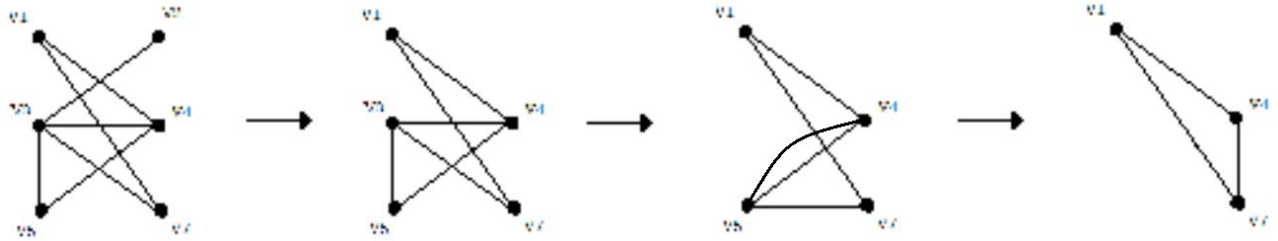


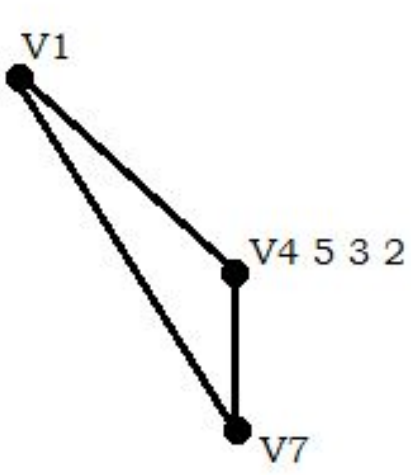
3) кільцева сума G1 та G2 (G1⊕G2) **4)** розщеплення вершини(V2) в G2



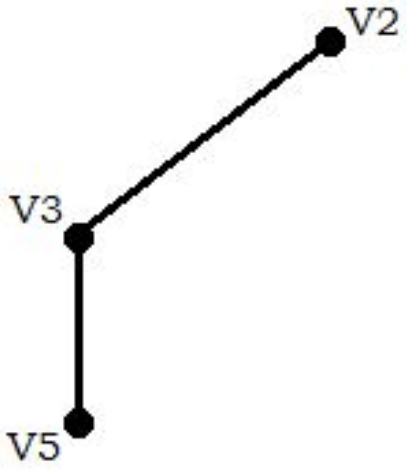


5) виділення підграфа A, що складається з 3-х вершин в G1 і стягнення A в G1 (G1\ A),



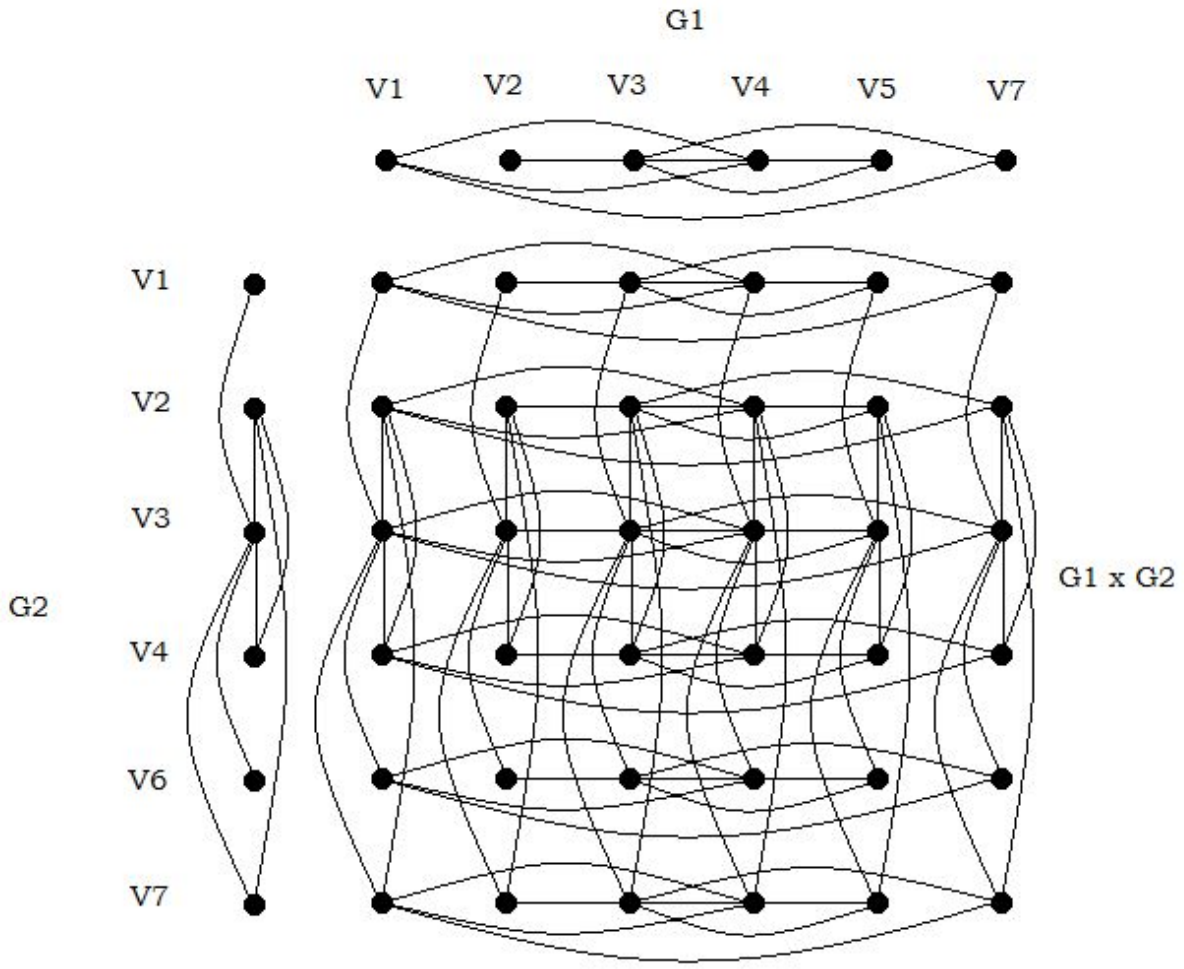


Підграф A

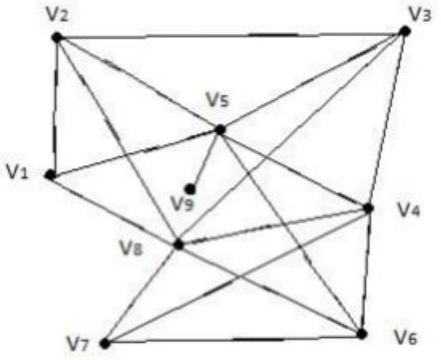


Стягнення A в G1 (G1 \ A)

6) добуток графів



Завдання № 2. Скласти таблицю суміжності для графа.



Таблиця суміжності:

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

Завдання № 3.

Для графа з другого завдання знайти діаметр.

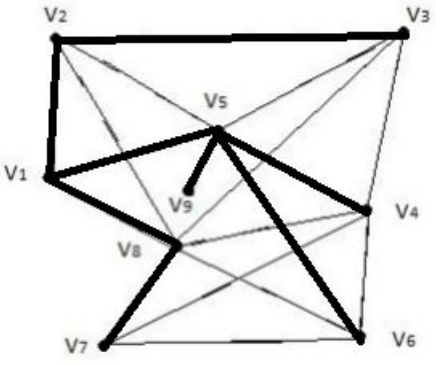
Діаметр = 3.

Завдання № 4.

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

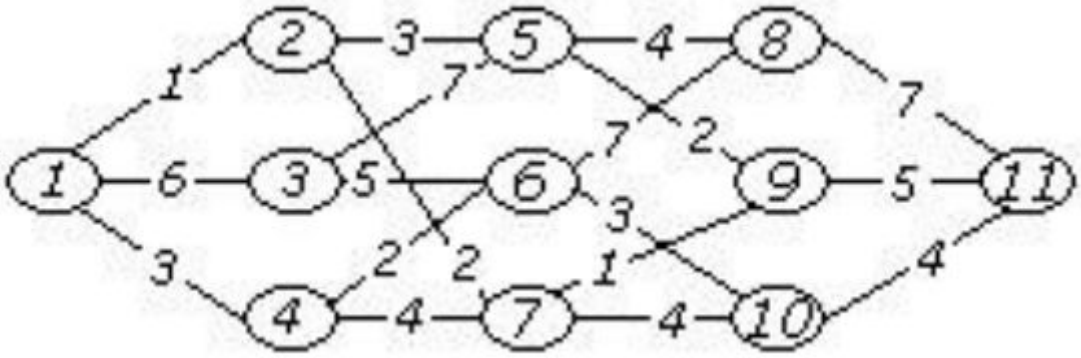
Вершина	BFS номер	Вміст черги
V1	1	V1
V2	2	V1V2
V5	3	V1V2V5
V8	4	V1V2V5V8
-	-	V2V5V8
V3	5	V2V5V8V3
-	-	V5V8V3
V4	6	V5V8V3V4
V6	7	V5V8V3V4V6
V9	8	V5V8V3V4V6V9
-	-	V8V3V4V6V9
V7	9	V8V3V4V6V9V7
-	-	V3V4V6V9V7
-	-	V4V6V9V7
-	-	V6V9V7
-	-	V9V7
-	-	V7
-	-	∅

Пошук вшир

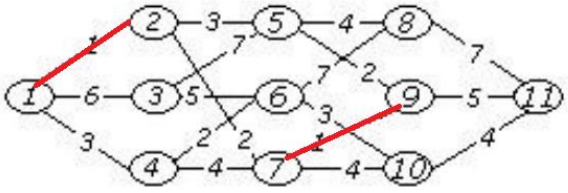


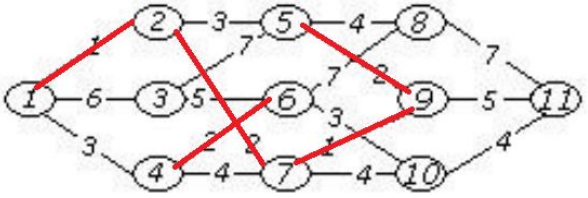
Завдання № 5.

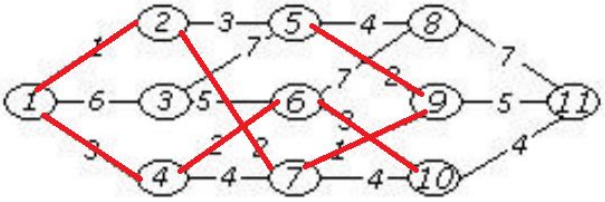
Знайти двома методами (Краскала і Прима) мінімальне остове дерево графа.

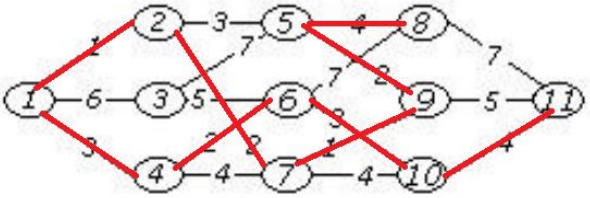


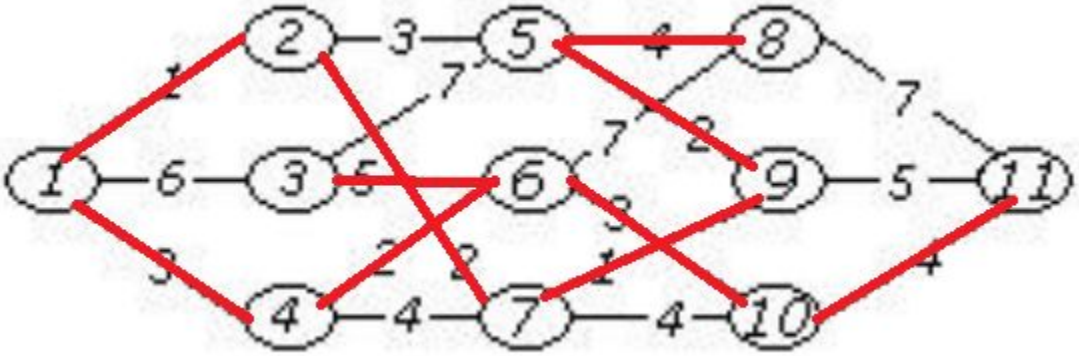
Метод Краскала:



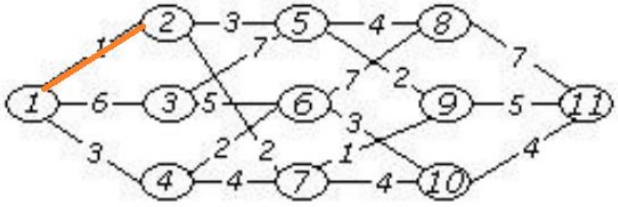


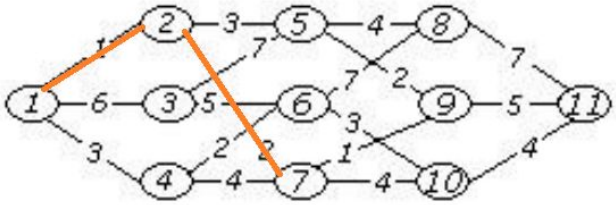


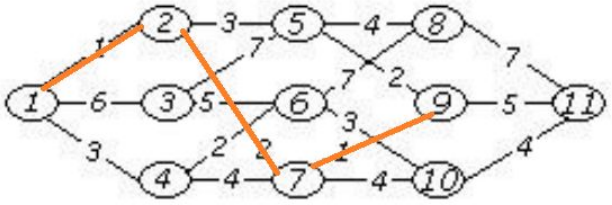


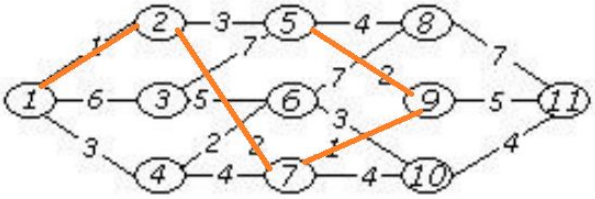


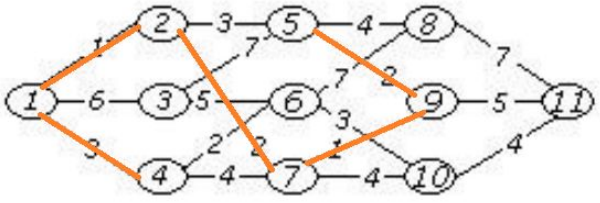
Метод Прима:

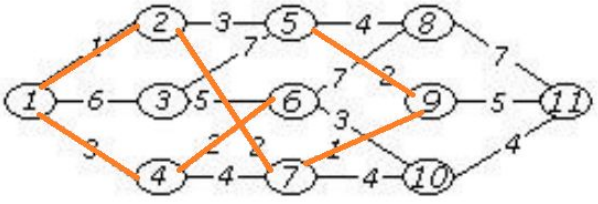


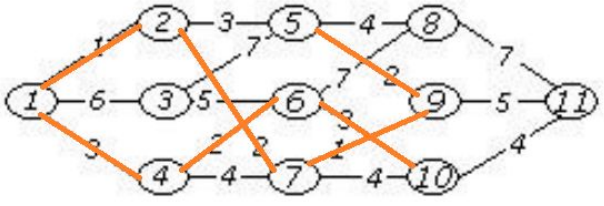


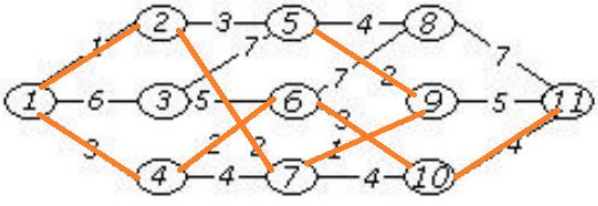


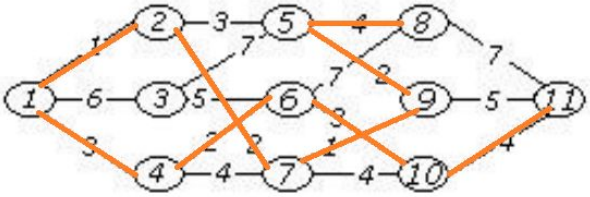


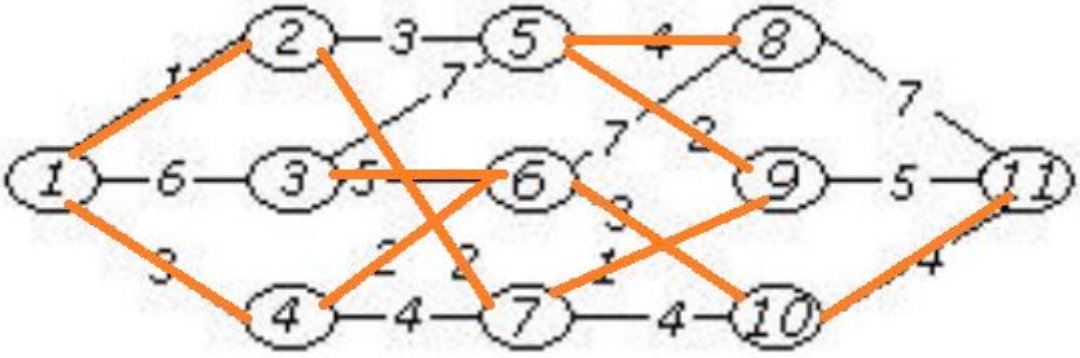








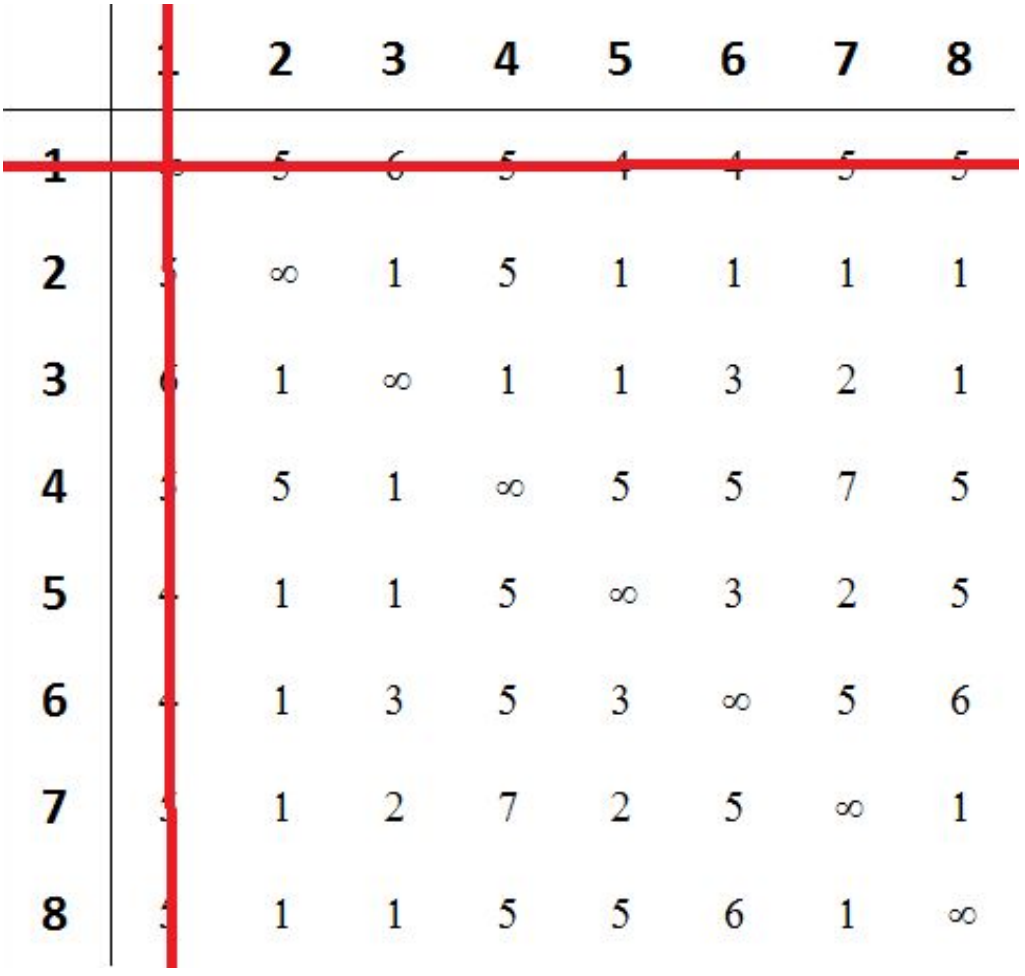

2
3
5
4
8
7
1
7
7
2
6
3
5
6
9
5
11
3
2
2
1
4
4
7
4
10


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

Завдання № 6

Розв’язати задачу комівояжера для повного 8-ми вершинного графа методом «іди у найближчий», матриця вагів якого має вигляд:

	1	2	3	4	5	6	7	8
1	∞	5	6	5	4	4	5	5
2	5	∞	1	5	1	1	1	1
3	6	1	∞	1	1	3	2	1
4	5	5	1	∞	5	5	7	5
5	4	1	1	5	∞	3	2	5
6	4	1	3	5	3	∞	5	6
7	5	1	2	7	2	5	∞	1
8	5	1	1	5	5	6	1	∞



	2	3	4	5	16	7	8
2	∞	1	5	1	1	1	1
3	1	∞	1	1	1	2	1
4	5	1	∞	5	5	7	5
5	1	1	5	∞	5	2	5
16	1	3	5	3	6	5	6
7	1	2	7	2	5	∞	1
8	1	1	5	5	5	1	∞

	162	3	4	5	7	8
162	0	1	5	1	1	1
3	1	∞	1	1	2	1
4	1	1	∞	5	7	5
5	1	1	5	∞	2	5
7	1	2	7	2	∞	1
8	1	1	5	5	1	∞

	3	4	5	7	1628
3	∞	1	1	2	
4	1	∞	5	7	
5	1	5	∞	2	
7	2	7	2	∞	
1628	1	5	5	1	

	3	4	5	16287
3	∞	1	1	
4	1	∞	5	
5	1	5	∞	
16287	2	7	2	

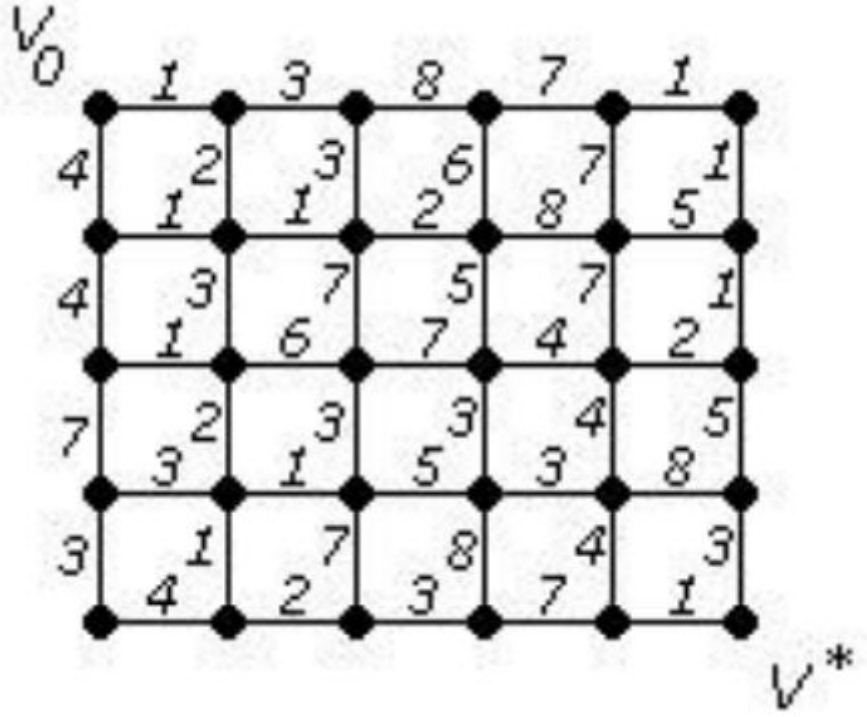
	3	4	162875
3	∞	1	
4	1	∞	
162875	1	5	

	4	1628753
4	∞	1
1628753	1	∞

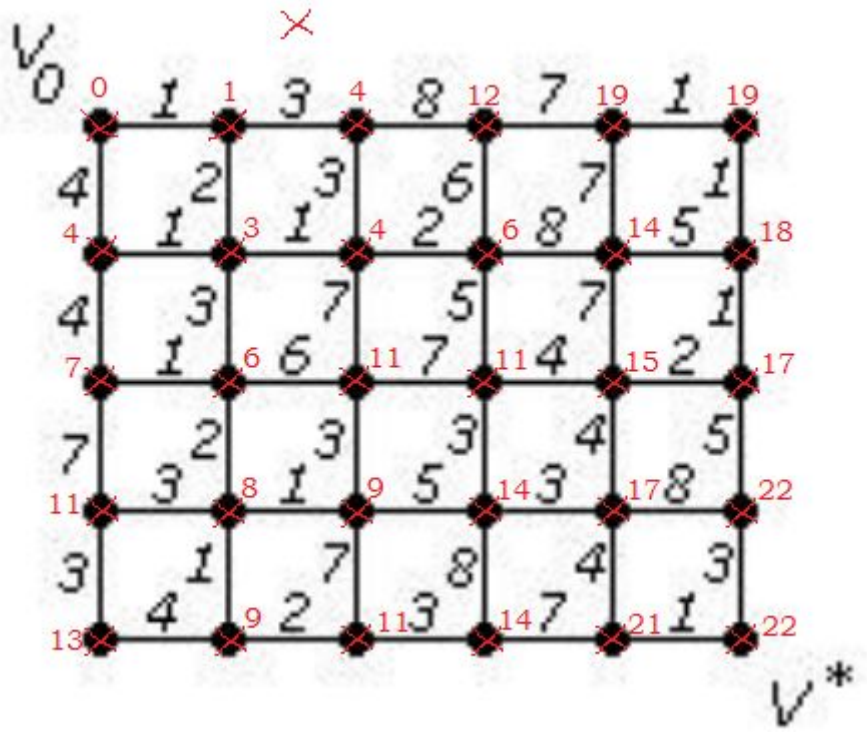
Шлях з мінімальною вагою – 1,6,2,8,7,5,3,4

Завдання № 7

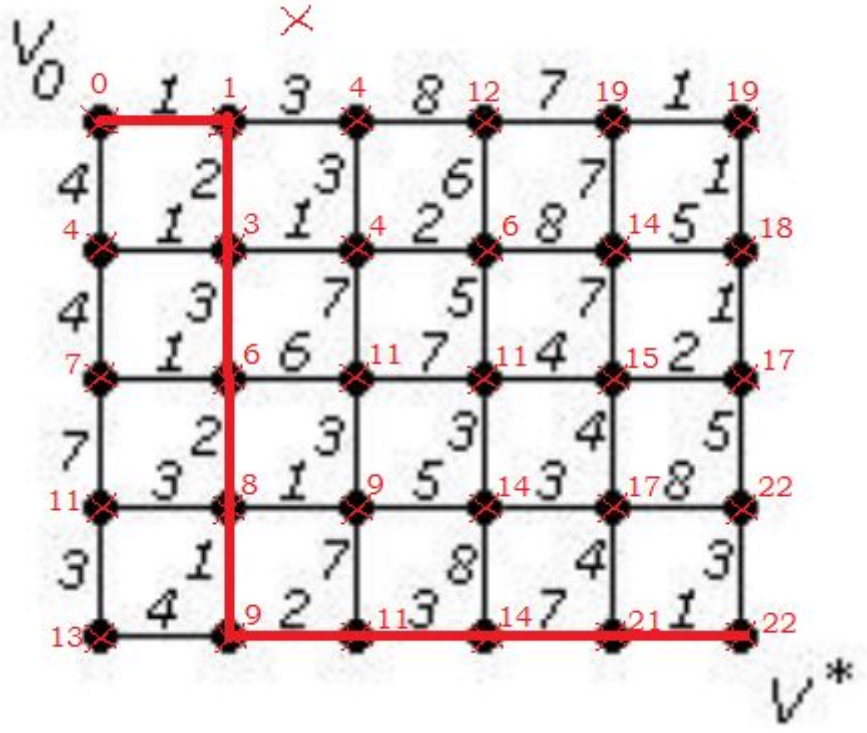
За допомогою алгоритму Дейкстри знайти найкоротший шлях у графі між парою вершин V_0 і V* .



За методом Дейкстри проходимо всі вершини і знаходимо вагу шляху до них



Далі віднімаючи від ваги вершини ваги сусідніх ребер знаходимо найкоротший шлях від V* до V_0



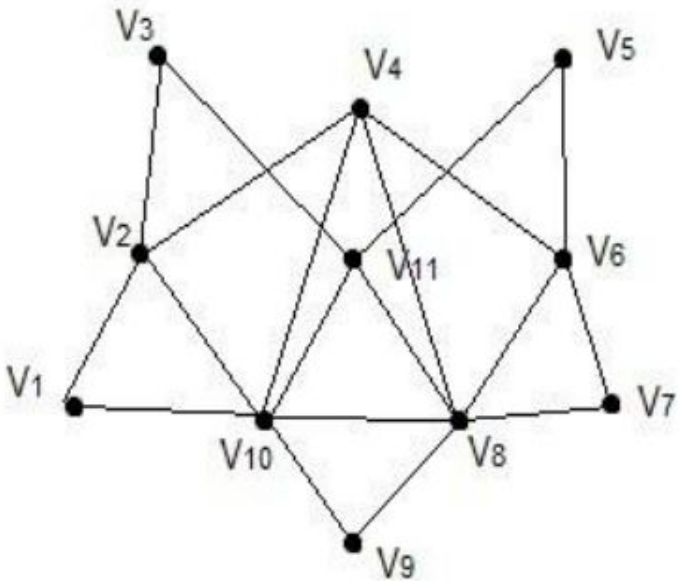
Найкоротший шлях від V* до V_0 (22)

Завдання № 8

Знайти ейлеровий цикл в ейлеровому графі двома методами:

а) Флері;

б) елементарних циклів.



а) Для знаходження ейлерового циклу почергово видаляю ребра перевіряючи, чи не є воно мостом.

Шлях ейлерового циклу за методом Флері:

V4, V6, V5, V11, V8, V7, V6, V8, V9, V10, V8, V4, V2, V1, V10, V11, V3, V2, V10, V4

б) Для знаходження ейлерового циклу виділяю елементарні цикли та поєдную їх в один

Шлях ейлерового циклу за методом елементарних циклів:

V4, V10, V2, V4, V8, V10, V9, V8, V6, V7, V8, V3, V2, V1, V10, V5, V6, V4

Завдання №9

Спростити формули (привести їх до скороченої ДНФ)

$$\overline{x}y \vee x\overline{yz}$$

Z \ X Y	0 0	0 1	11	10
0	0	1	I 0	II 1
1	0	1	0	0

I - $\bar{x}y$

II - $x\bar{y}\bar{z}$

$$\bar{x}y \vee x\bar{y}\bar{z} \text{ (ДНФ)}$$

Мінімізувати функцію неможливо

Програмно реалізовані алгоритми:

1)Обхід вглиб

```
#include <iostream>

using namespace std;

bool isUsed(int arr[], int size, int n) {
    bool t = 0;
    for (int x = 0; x < size; x++) {
        if (arr[x] == n)
            t = 1;
    }
    return t;
}

int main() {
    int vertices [8]{0};

    int buffer [8]{0};

    int c_a = 0;
    int c_b = 0;

    bool x[8][8] = {
            {0,1,1,0,0,0,0,0},
            {1,0,0,1,0,0,0,0},
            {1,0,0,1,0,0,0,0},
            {0,1,1,0,1,0,0,0},
```

```
        {0,0,0,1,0,1,0,1},
        {0,0,0,0,1,0,1,0},
        {0,0,0,0,0,1,0,0},
        {0,0,0,0,1,0,0,0}};

    bool add = 0;

    for (int a = 0; a < c_b+1; a++) {
        cout << buffer[a]+1 << " ";
    }
    cout << "\n";

    while(c_a < 7){
        add = 0;

        for(int i = 0; i < 8; i++) {
            if (x[buffer[c_b]][i] == 1 && !isUsed(vertices, 8, i)) {
                add = 1;
                vertices[++c_a] = i;
                buffer[++c_b] = i;
                for (int a = 0; a < c_b + 1; a++) {
                    cout << buffer[a] + 1 << " ";
                }
                cout << "\n";
                break;
            }
        }

        if (c_b != 0 && !add) {
            buffer[c_b--] = 0;

            for (int a = 0; a < c_b+1; a++) {
                cout << buffer[a]+1 << " ";
            }
            cout << "\n";
        }
    }

    while (c_b > 0){
        buffer[c_b--] = 0;

        for (int a = 0; a < c_b+1; a++) {
            cout << buffer[a]+1 << " ";
        }
        cout << "\n";
    }
}
```

Результат:

```
/Users/bogdankalysh/CLionProjects/Rozraha_1/cmake-build-debug/Rozraha_1
1
1 2
1 2 4
1 2 4 3
1 2 4
1 2 4 5
1 2 4 5 6
1 2 4 5 6 7
1 2 4 5 6
1 2 4 5
1 2 4 5 8
1 2 4 5
1 2 4
1 2
1

Process finished with exit code 0
```

2)Обхід вшир

```
#include <iostream>

using namespace std;

bool isUsed(int arr[], int size, int n) {
    bool t = 0;
    for (int x = 0; x < size; x++) {
        if (arr[x] == n)
            t = 1;
    }
    return t;
}

int main() {
    bool x[8][8] = {
            {0,1,1,0,0,0,0,0},
            {1,0,0,1,0,0,0,0},
            {1,0,0,1,0,0,0,0},
            {0,1,1,0,1,0,0,0},
            {0,0,0,1,0,1,0,1},
            {0,0,0,0,1,0,1,0},
            {0,0,0,0,0,1,0,0},
            {0,0,0,0,1,0,0,0}};

    int vertices [8]{0};
    int buffer [8]{0};

    int c_a = 0;
    int c_b = 0;

    bool add;

    while(c_a < 7){
        add = 0;

        for(int i = 0; i < 8; i++) {
            if(x[buffer[0]][i] == 1 && !isUsed(vertices, 8, i)) {
                add = 1;
                vertices[++c_a] = i;
                buffer[++c_b] = i;
                for (int a = 0; a < c_b + 1; a++) {
                    cout << buffer[a] + 1 << " ";
                }
                cout << "\n";
                break;
            }
        }

        if (!add) {
            for(int a = 0; a < c_b+1; a++) {
                buffer[a] = buffer[a+1];
                if(a >= c_b){
                    buffer[a] = 0;
                }
            }
            c_b--;

            for (int a = 0; a < c_b+1; a++) {
                cout << buffer[a]+1 << " ";
            }
            cout << "\n";
        }
    }

    while (c_b > 0){
        for(int a = 0; a < c_b+1; a++) {
            buffer[a] = buffer[a+1];
            if(a >= c_b){
                buffer[a] = 0;
```

```
        }
    }
    c_b--;
    for (int a = 0; a < c_b+1; a++) {
        cout << buffer[a]+1 << " ";
    }
    cout << "\n";
  }
}
```

Результат:

```
/Users/bogdankalysh/CLionProjects/Rozraha_2/cmake-build-debug/Rozraha_2
1 2
1 2 3
2 3
2 3 4
3 4
4
4 5
5
5 6
5 6 8
6 8
6 8 7
8 7
7

Process finished with exit code 0
```

3)Алгоритм Прима

```cpp
#include <iostream>

using namespace std;

int main() {

    int tree [11][11] {
            {0,4,3,2,0,0,0,0,0,0,0},
            {0,0,0,0,2,0,1,0,0,0,0},
            {0,0,0,0,6,7,0,0,0,0,0},
            {0,0,0,0,0,2,4,0,0,0,0},
            {0,2,6,0,0,0,0,7,5,0,0},
            {0,0,7,2,0,0,0,4,0,3,0},
            {0,1,0,4,0,0,0,0,4,5,0},
            {0,0,0,0,7,4,0,0,0,0,7},
            {0,0,0,0,5,0,4,0,0,0,1},
            {0,0,0,0,0,3,5,0,0,0,3},
            {0,0,0,0,0,0,0,7,1,3,0},
    };
    // записую матрицю інцидентності

    int min(100), min_e, min_x, count(1);
    int edges[11];
    edges[0] = 0;

    cout << 1 << endl;

    do {

        for(int i = 0; i < count; i++) {
            for (int x = 0; x < 11; x++) {
                if (tree[edges[i]][x] != 0 && tree[edges[i]][x] < min) {
                    min = tree[edges[i]][x];
                    min_e = x;
                    min_x = edges[i];
                }
            }
        }
        //знаходжу мінімальне ребро серед вершин, які відкриті

        edges[count] = min_e;
        // додаю до масиву наступну вершину

        count++;
        min = 10;

        for (int a = 0; a < 11; a++) {
            tree[a][min_e] = 0;
        }
        // занулюю стовбчик ребер з відкритою вершиною

        cout << min_x+1 << "-->" << min_e+1 << "\t(" << min_e + 1 << ")" << endl;

    }while (count < 11);
}
```

Результат:

```
/Users/bogdankalysh/CLionProjects/Rozraha_3/cmake-build-debug/Rozraha_3
1
1-->4   (4)
4-->6   (6)
1-->3   (3)
6-->10  (10)
10-->11 (11)
11-->9  (9)
1-->2   (2)
2-->7   (7)
2-->5   (5)
6-->8   (8)

Process finished with exit code 0
```

4)Алгоритм Краскала

```
#include <stdio.h>
#include <iostream>
#include <stdlib.h>
using namespace std;

const int q = 11;
int BuildTrees(int n, int A[q][q]);
void DeleteDuplicates(int n, int A[q][q]);
int InDifferTrees(int n, int A[q][q], int first, int second);
void AddToTheTree(int n, int A[q][q], int first, int second);

int main()
{
    setlocale(LC_ALL, "Ukrainian");
    int A[11][11] =
            { 0, 2, 5, 7, 0, 0, 0, 0, 0, 0, 0,
              2, 0, 0, 0, 7, 0, 7, 0, 0, 0, 0,
              5, 0, 0, 0, 5, 4, 0, 0, 0, 0, 0,
              7, 0, 0, 0, 0, 3, 1, 0, 0, 0, 0,
              0, 7, 5, 0, 0, 0, 0, 4, 1, 0, 0,
              0, 0, 4, 3, 0, 0, 0, 4, 0, 2, 0,
              0, 7, 0, 1, 0, 0, 0, 0, 3, 2, 0,
              0, 0, 0, 0, 4, 4, 0, 0, 0, 0, 3,
              0, 0, 0, 0, 1, 0, 3, 0, 0, 0, 6,
              0, 0, 0, 0, 0, 2, 2, 0, 0, 0, 4,
              0, 0, 0, 0, 0, 0, 0, 3, 6, 4, 0 };

    DeleteDuplicates(11, A);
    for (int i = 1; i <= 7; i++){
        cout << "\nВузли з вагою: " << i << ": ";
        for (int j = 1; j <= 11; j++){
            for (int k = 1; k <= 11; k++){
                if (A[j - 1][k - 1] == i){
                    cout << " " << j << "-" << k;;
                }
            }
        }
    }
    cout << "\n";

    int B[11][11];
    BuildTrees(11, B);
    cout << "\n\nНове дерево: ";//вага 7 - максимальна вага
    for (int i = 1; i <= 7; i++){
        //перший вузол
        for (int j = 1; j <= 11; j++){
            //другий вузол
            for (int k = 1; k <= 11; k++){
                if (A[j - 1][k - 1] == i && InDifferTrees(11, B, j, k)){
                    AddToTheTree(11, B, j, k);
                    cout << " " << j << "-" << k;
                }
            }
        }
    }
    return 0;
}
```

```
void DeleteDuplicates(int n, int A[q][q]) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (j < i) {
                A[i][j] = 0;
            }
        }
    }
}

int BuildTrees(int n, int A[q][q]) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            A[i][j] = 0;
        }
    }
    for (int i = 0; i < n; i++) {
        A[i][i] = i + 1;
    }
    return A[n][n];
}
//Перевірте відсортовані вузли та додайте до дерева

void AddToTheTree(int n, int A[q][q], int first, int second) {
    int scndLine;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (A[i][j] == second) {
                scndLine = i;
            }
        }
    }
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (A[i][j] == first) {
                for (int k = 0; k < n; k++) {
                    if (A[scndLine][k]) {
                        A[i][k] = A[scndLine][k];
                        A[scndLine][k] = 0;
                    }
                }
            }
        }
    }
}

int InDifferTrees(int n, int A[q][q], int first, int second){
    int temp1, temp2;
    //Лінія
    for (int i = 0; i < n; i++){
        temp1 = 0;
        temp2 = 0;
        //перший елемент
        for (int j = 0; j < n; j++){
            if (A[i][j] == first){
                temp1 = 1;
            }
        }
        //другий елемент
        for (int k = 0; k < n; k++){
            if (A[i][k] == second){
                temp2 = 1;
            }
        }
        if (temp1 && temp2){
            return 0;
        }
    }
    return 1;
}
```

Результат:

```
/Users/bogdankalysh/CLionProjects/Rozraha_4/cmake-build-debug/Rozraha_4

Вузли з вагою: 1:  4-7 5-9
Вузли з вагою: 2:  1-2 6-10 7-10
Вузли з вагою: 3:  4-6 7-9 8-11
Вузли з вагою: 4:  3-6 5-8 6-8 10-11
Вузли з вагою: 5:  1-3 3-5
Вузли з вагою: 6:  9-11
Вузли з вагою: 7:  1-4 2-5 2-7


Нове дерево:  4-7 5-9 1-2 6-10 7-10 7-9 8-11 3-6 5-8 1-3
Process finished with exit code 0
```

5)Алгоритм Дейкстри

```
#include <iostream>
using namespace std;

int main()
{
    setlocale(LC_ALL,"Ukrainian");
    int n,i,j;
    n = 30;//кількість вершин
    int am[n][n];
    for (i=0;i<n;i++)
    {
        for (j=0;j<n;j++){
            am[i][j]=0;
        }
    }

    am[0][1]=am[1][0]=6;
    am[0][6]=am[6][0]=4;
    am[1][2]=am[2][1]=1;
    am[1][7]=am[7][1]=8;
    am[2][3]=am[3][2]=1;
    am[2][8]=am[8][2]=3;
    am[3][4]=am[4][3]=3;
    am[3][9]=am[9][3]=1;
    am[4][5]=am[5][4]=3;
    am[4][10]=am[10][4]=5;
    am[5][11]=am[11][5]=7;
    am[6][7]=am[7][6]=2;
    am[6][12]=am[12][6]=5;
    am[7][8]=am[8][7]=1;
    am[7][13]=am[13][7]=1;
    am[8][9]=am[9][8]=4;
    am[8][14]=am[14][8]=3;
    am[9][10]=am[10][9]=2;
    am[9][15]=am[15][9]=4;
    am[10][11]=am[11][10]=4;
    am[10][16]=am[16][10]=1;
    am[11][17]=am[17][11]=7;
    am[12][13]=am[13][12]=7;
    am[12][18]=am[18][12]=5;
    am[13][14]=am[14][13]=1;
    am[13][19]=am[19][13]=7;
    am[14][15]=am[15][14]=2;
    am[14][20]=am[20][14]=1;
    am[15][16]=am[16][15]=3;
    am[15][21]=am[21][15]=4;
    am[16][17]=am[17][16]=7;
    am[16][22]=am[22][16]=2;
    am[17][23]=am[23][17]=8;
    am[18][19]=am[19][18]=7;
    am[18][24]=am[24][18]=8;
    am[19][20]=am[20][19]=3;
    am[19][25]=am[25][19]=2;
    am[20][21]=am[21][20]=1;
    am[20][26]=am[26][20]=1;
    am[21][22]=am[22][21]=8;
    am[21][27]=am[27][21]=3;
    am[22][28]=am[28][22]=3;
    am[22][23]=am[23][22]=5;
    am[23][29]=am[29][23]=7;
    am[24][25]=am[25][24]=4;
    am[25][26]=am[26][25]=7;
    am[26][27]=am[27][26]=3;
    am[27][28]=am[28][27]=3;
    am[28][29]=am[29][28]=6;

    int numb[30]{-1};
    int rebra=0;
    for (i=0;i<n;i++)
```

```
    {
        for (j=0;j<n;j++)
        {
            if (am[i][j]!=0)
                rebra++;// кількість ребер
        }
    }
    int weight[n]; //ваги
    bool visited[n]; //пройдені
    for (i=0;i<n;i++)
    {
        weight[i]=10000000;
        visited[i]=0;
    }
    weight[0]=0;
    visited[0]=1;
    int nmin,Vmin1,Vmin2;
    while (rebra!=0)
    {
        nmin=10000000;
        for (i=0;i<n;i++)
        {
            if (visited[i]==1)//якщо була пройдена
            {
                for (j=0;j<n;j++)
                {
                    if (weight[i]+am[i][j]<nmin&&am[i][j]!=0)//по рядку і шукає мін
                    {
                        nmin=weight[i]+am[i][j];
                        Vmin1=i;
                        Vmin2=j;
                    }
                }
            }
        }
        if (weight[Vmin2]>nmin)
        {
            weight[Vmin2]=nmin;
            numb[Vmin2]=Vmin1;
        }
        visited[Vmin2]=1;
        am[Vmin1][Vmin2]=am[Vmin2][Vmin1]=0;
        rebra-=2;//віднімаю ребра
    }

    int endd=29;
    int way[n];
    i=0;
    cout<<"\nВага мінімального шляху = "<<weight[endd]<<endl;

    cout<<"\nШлях:"<<endl;
    while (endd!=0)
    {
        way[i]=endd;
        endd=numb[endd];
        i++;
    }
    way[i]=0;

    for (i;i>=0;i--)
    {
        cout<<way[i];
        if (i!=0) cout<<"->";
    }
    cout<<endl;
}
```

Результат:

```
/Users/bogdankalysh/CLionProjects/Laba_DM_5/cmake-build-debug/Laba_DM_5

Вага мінімального шляху = 22

Шлях:
0->6->7->13->14->20->21->27->28->29

Process finished with exit code 0
```

6)Алгоритм “Іди в найближчий” для розв’язання задачі комівояжера.

```
#include <iostream>

void print (int arr [8][8]) {
    for (int x = 0; x < 8; x++) {
        bool hugh = 0;
        for(int y = 0; y < 8; y++) {
            if(arr[x][y] != 0) {
                hugh = 1;
                std::cout << arr[x][y] << " ";
            }
        }
        if(hugh)
        std::cout << "\n";
    }
    std::cout << "\n";
}

int main() {
    int matrice[8][8] {
            {10,5,6,5,4,4,5,5},
            {5,10,1,5,1,1,1,1},
            {6,1,10,1,1,3,2,1},
            {5,5,1,10,5,5,7,5},
            {4,1,1,5,10,3,2,5},
            {4,1,3,5,3,10,5,6},
            {5,1,2,7,2,5,10,1},
            {5,1,1,5,5,6,1,10},
    };

    int nums [8] {0};
    nums[0] = 0;
    int count = 1;
    int path(0);

    int min_num, min_el;

    while (count < 8) {
        min_num = 10;

        print(matrice);

        for (int x = 0; x < 8; x++) {
            if(matrice [nums[count-1]][x] <= min_num && matrice [nums[count-1]][x] != 0) {
                min_num = matrice [nums[count-1]][x];
                min_el = x;
            }
        }
        nums[count] = min_el;
        path += min_num;

        for (int x = 0; x < 8; x++) {
            matrice[nums[count-1]][x] = 0;
            matrice[x][nums[count-1]] = 0;
        }

        count++;
    }

    for (int i = 0; i < 8; i++) {
        std::cout << nums[i]+1 << "\t";
    }
}
```

```
/Users/bogdankalysh/CLionProjects/Rozraha_6/cmake-build-debug/Rozraha_6
10 5 6 5 4 4 5 5
5 10 1 5 1 1 1 1
6 1 10 1 1 3 2 1
5 5 1 10 5 5 7 5
4 1 1 5 10 3 2 5
4 1 3 5 3 10 5 6
5 1 2 7 2 5 10 1
5 1 1 5 5 6 1 10

10 1 5 1 1 1 1
1 10 1 1 3 2 1
5 1 10 5 5 7 5
1 1 5 10 3 2 5
1 3 5 3 10 5 6
1 2 7 2 5 10 1
1 1 5 5 6 1 10

10 1 5 1 1 1
1 10 1 1 2 1
5 1 10 5 7 5
1 1 5 10 2 5
1 2 7 2 10 1
1 1 5 5 1 10

10 1 1 2 1
1 10 5 7 5
1 5 10 2 5
2 7 2 10 1
1 5 5 1 10

10 1 1 2
1 10 5 7
1 5 10 2
2 7 2 10

10 1 1
1 10 5
1 5 10

10 1
1 10
```

```
1   6   2   8   7   5   3   4
Process finished with exit code 0
```