МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота №5

з дисципліни

«Дискретна математика»

Виконав: студент групи
КН-113

ПІ студента: Сидорук Михайло
Викладач: Мельникова Н.І.

Львів 2019

Лабораторна робота № 5.

Тема: Знаходження найкоротшого маршруту за алгоритмом Дейкстри. Плоскі планарні графи

Мета роботи: набуття практичних вмінь та навичок з використання алгоритму Дейкстри.

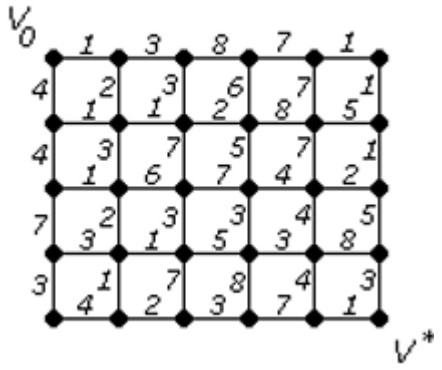
Варіант 12

Завдання № 1.

Розв'язати на графах наступні 2 задачі:

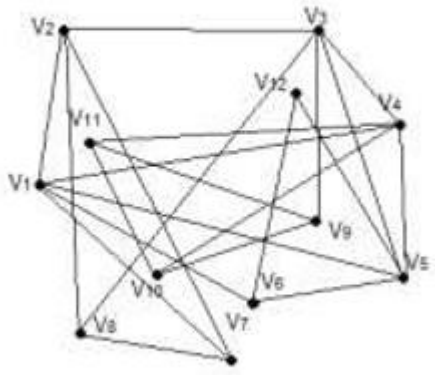
1. За допомогою алгоритму Дейкстра знайти найкоротший шлях у графі поміж парою вершин V0 і V* .

12



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

12



Розв’язання

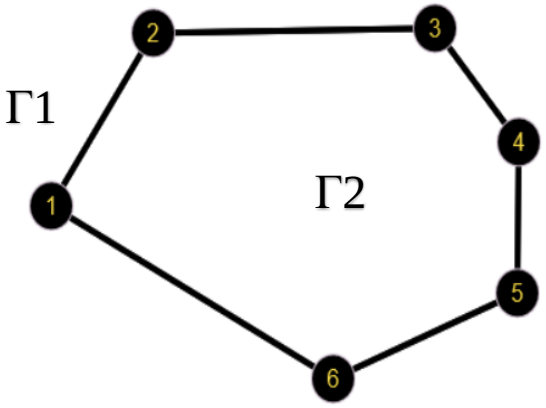
1.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	-	1	∞	∞	∞	∞	4	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	-	-	4	∞	∞	∞	4	3	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
3	-	-	4	∞	∞	∞	4	-	4	∞	∞	∞	∞	6	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
4	-	-	-	12	∞	∞	4	-	4	∞	∞	∞	∞	6	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
5	-	-	-	12	∞	∞	-	-	4	∞	∞	∞	8	6	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
6	-	-	-	12	∞	∞	-	-	-	6	∞	∞	8	6	11	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
7	-	-	-	12	∞	∞	-	-	-	-	14	∞	8	6	11	11	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
8	-	-	-	12	∞	∞	-	-	-	-	14	∞	7	-	11	11	∞	∞	∞	8	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
9	-	-	-	12	∞	∞	-	-	-	-	14	∞	-	-	11	11	∞	∞	14	8	14	∞	∞	∞	∞	∞	∞	∞	∞	∞
10	-	-	-	12	∞	∞	-	-	-	-	14	∞	-	-	11	11	∞	∞	11	-	9	∞	∞	∞	∞	9	∞	∞	∞	∞
11	-	-	-	12	∞	∞	-	-	-	-	14	∞	-	-	11	11	∞	∞	11	-	-	14	∞	∞	∞	9	16	∞	∞	∞
12	-	-	-	12	∞	∞	-	-	-	-	14	∞	-	-	11	11	∞	∞	11	-	-	14	∞	∞	13	-	11	∞	∞	∞
13	-	-	-	12	∞	∞	-	-	-	-	14	∞	-	-	-	11	∞	∞	11	-	-	14	∞	∞	13	-	11	∞	∞	∞
14	-	-	-	12	∞	∞	-	-	-	-	14	∞	-	-	-	-	15	∞	11	-	-	14	∞	∞	13	-	11	∞	∞	∞
15	-	-	-	12	∞	∞	-	-	-	-	14	∞	-	-	-	-	15	∞	-	-	-	14	∞	∞	13	-	11	∞	∞	∞
16	-	-	-	12	∞	∞	-	-	-	-	14	∞	-	-	-	-	15	∞	-	-	-	14	∞	∞	13	-	-	18	∞	∞
17	-	-	-	-	19	∞	-	-	-	-	14	∞	-	-	-	-	15	∞	-	-	-	14	∞	∞	13	-	-	18	∞	∞
18	-	-	-	-	19	∞	-	-	-	-	14	∞	-	-	-	-	15	∞	-	-	-	14	∞	∞	-	-	-	18	∞	∞
19	-	-	-	-	19	∞	-	-	-	-	-	19	-	-	-	-	15	∞	-	-	-	14	∞	∞	-	-	-	18	∞	∞
20	-	-	-	-	19	∞	-	-	-	-	-	19	-	-	-	-	15	∞	-	-	-	-	17	∞	-	-	-	18	∞	∞
21	-	-	-	-	19	∞	-	-	-	-	-	19	-	-	-	-	-	17	-	-	-	-	17	∞	-	-	-	18	∞	∞
22	-	-	-	-	19	∞	-	-	-	-	-	19	-	-	-	-	-	-	-	-	-	-	17	23	-	-	-	18	∞	∞
23	-	-	-	-	19	∞	-	-	-	-	-	19	-	-	-	-	-	-	-	-	-	-	-	23	-	-	-	18	21	∞
24	-	-	-	-	19	∞	-	-	-	-	-	19	-	-	-	-	-	-	-	-	-	-	-	23	-	-	-	-	21	∞
25	-	-	-	-	-	20	-	-	-	-	-	19	-	-	-	-	-	-	-	-	-	-	-	23	-	-	-	-	21	∞
26	-	-	-	-	-	20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	23	-	-	-	-	21	∞
27	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	23	-	-	-	-	21	∞
28	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	23	-	-	-	-	-	22
29	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	23	-	-	-	-	-	-
30	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

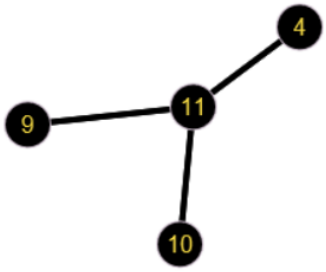
Мінімальний шлях до вершини 30 – 22.

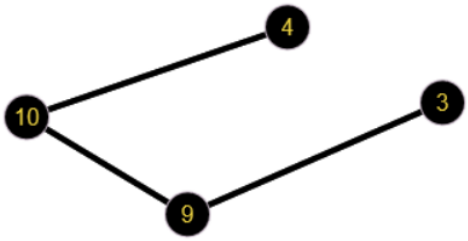
2.

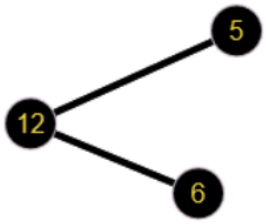
Вибраний початковий цикл:

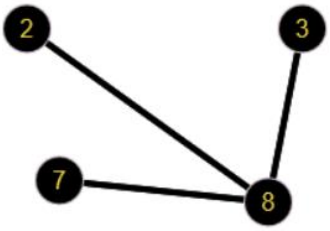


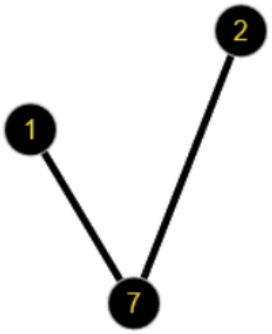
Сегменти графа:



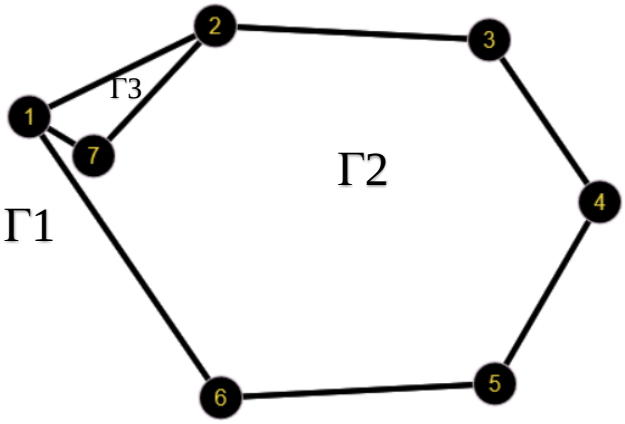


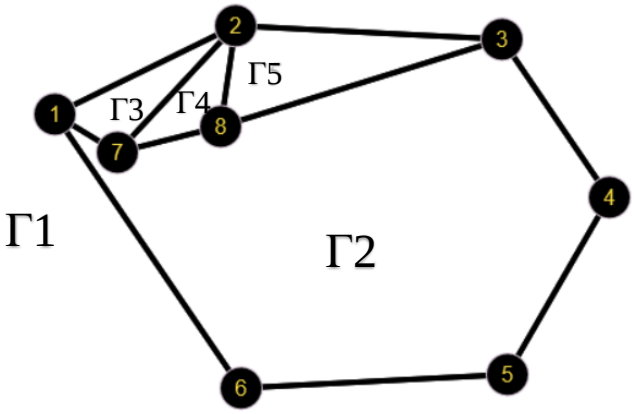


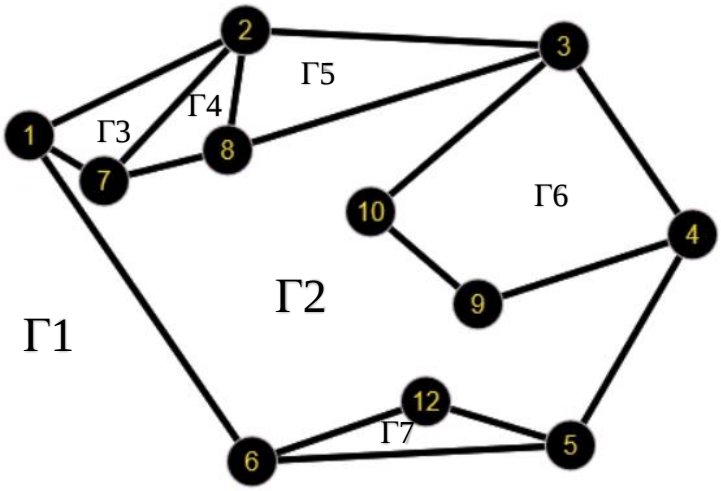


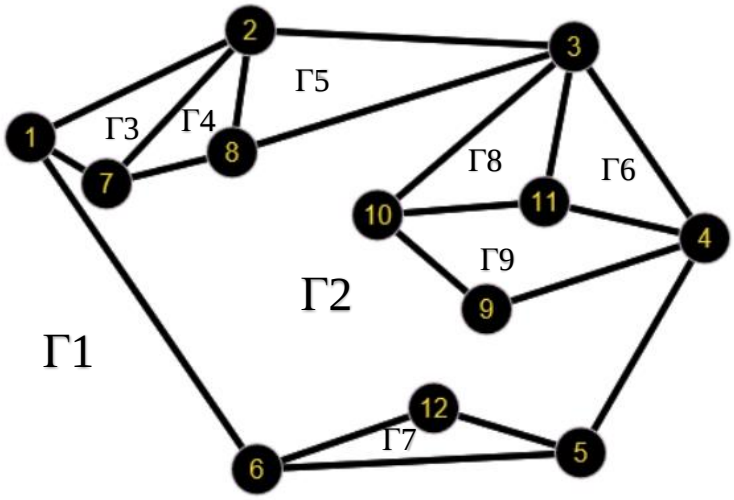


Порядок включення сегментів:


2
3
Г3
1
7
Г2
4
Г1
6
5


2
3
Г5
Г3
Г4
1
8
7
4
Г1
Г2
6
5

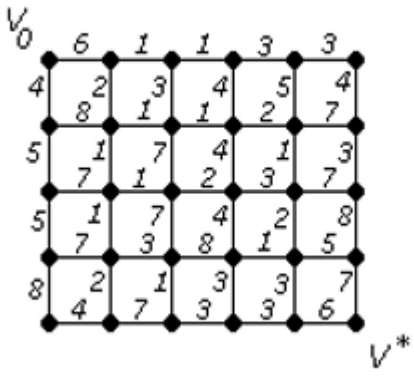

2
3
Г5
Г4
Г3
1
8
7
Г6
10
4
Г2
9
Г1
12
Г7
6
5



Завдання №2.

Написати програму, яка реалізує алгоритм Дейкстри знаходження найкоротшого шляху між парою вершин у графі. Протестувати розроблену програму на графі згідно свого варіанту.

12



Розв'язок:

```
#include "pch.h"
#include <iostream>
#include <sstream>
#include <fstream>
#include <stdlib.h>
using namespace std;
string path = "MyFile.txt";
ifstream fin;
ofstream fout;

struct vershina {
```

```
        int number;
        bool proid = false;
        bool wiev = false;
        int minleng = 1000;
        string way="";
};
struct rebro {
        int v1;
        int v2;
        int leng;
};
void Add(rebro *reb, int i) {
        setlocale(LC_ALL, "Ukrainian");
        string str;


        /*cout <<"Введіть вагу " << i + 1 << " ребра: " ;
        cin >> reb[i].leng;
        cout <<"Перша суміжна вершина: ";
        cin >> reb[i].v1;
        cout << "Друга суміжна вершина: ";
        cin >> reb[i].v2;
        cout << endl;*/
        str = "";
        getline(fin, str);
        reb[i].leng = atoi(str.c_str());

        str = "";
        getline(fin, str);
         reb[i].v1=atoi(str.c_str());

        str = "";
        getline(fin, str);
         reb[i].v2=atoi(str.c_str());

}


int main() {
        setlocale(LC_ALL, "Ukrainian");

        int n = 49, m = 0;
        int k = 0;
        int x = 0;
        int min;
        int minleng = 1000;
        stringstream ss[200];
        string str;
        int t = 0;
        int begin, end;
        /*cout << "Введіть кількість ребер у графі:  ";
        cin >> n;
        cout << endl;*/
        rebro *reb = new rebro[n];
        vershina v[30];
        cout << "З якої вершини почати? (не більше 30) ";
        cin >> begin;
        cout << "До якої вершини знайти шлях?(не більше 30) ";
        cin >> end;
```

```
for (int i = 0; i < 30; i++)
{
	v[i].number = i+1;
}

fin.open(path);
for (int i = 0; i < n; i++)
{
	Add(reb, i);
}
fin.close();
v[begin-1].wiev = true;
v[begin-1].minleng = 0;
ss[t] << begin;
ss[t] >> str;
v[begin-1].way += str;
t++;
str = "";

while (v[end-1].proid == false )
{
	for(int i=0; i<30; i++)
	{
		if(v[i].wiev==true && v[i].proid==false)
		{
			//cout << v[i].number<<" "<<" вершина з шляхом не 1000"<<" ";
			m++;
		}
	}
	//cout << endl;
	//кількість вершин до яких вже визначений шлях.
	//cout << m<<" кількість з шляхом не 1000"<<endl;

	int *mas_v = new int [m]; //виділення пам'яті під них.

	for (int i = 0; i < 30; i++)
	{
		if (v[i].wiev==true && v[i].proid==false)
		{
			mas_v[k] = i;
			//cout << mas_v[k] << " - index of wievs ";
			k++;
		}
	}
	//cout << endl;
	//заповнення масиву вершин, що можуть розглядатися.
	k = 0;



	for (int i = 0; i < 30; i++)
	{

		if (v[i].wiev == true && v[i].minleng<minleng && v[i].proid==false)
		{
			min = i;
			minleng = v[i].minleng;
```

```
	}



}
	//cout << v[min].number <<" індекс вершини з найменшим шляхом"<< endl;

//визначення вершини відстань до якої найменша.



for (int i = 0; i < n; i++)
{
	if (v[min].number == reb[i].v1 || v[min].number == reb[i].v2)
	{
		x++;
	}
}
//cout <<"суміжних непройдених вершин: " << x<<<endl;
//визначення кількості суміжих вершин.

int *mas_sum = new int[x];
//виділення пам'яті під суміжні вершини.

m = 0;
for (int i = 0; i < n; i++)
{
	if (v[min].number == reb[i].v1)
	{
		for (int j = 0; j < 30; j++)
		{
			if (v[j].number == reb[i].v2 && v[j].proid==false)
			{
				mas_sum[m] = j;

				v[j].wiev = true;

				//cout << v[j].number << " -індекс суміжної ";
				m++;

				break;
			}
		}
	}
	if (v[min].number == reb[i].v2)
	{
		for (int j = 0; j < 30; j++)
		{
			if (v[j].number == reb[i].v1&& v[j].proid == false)
			{
				mas_sum[m] = j;

				v[j].wiev = true;

				//cout << v[j].number << "-індекс суміжної ";
				m++;
			}
		}
	}
}
//заповнення масиву суміжних вершин.
```

```
		for (int i = 0; i < m; i++)
		{
			for (int j = 0; j < n; j++)
			{

				if ((reb[j].v1 == v[min].number || reb[j].v2 ==v[min].number) &&
(reb[j].v1 == v[mas_sum[i]].number || reb[j].v2 == v[mas_sum[i]].number))
				{
					//cout << i <<" "<<j<< endl;
					if (v[mas_sum[i]].minleng > (v[min].minleng +
reb[j].leng))
					{
						v[mas_sum[i]].minleng = v[min].minleng +
reb[j].leng;
						//cout << v[mas_sum[i]].number << "
"<<v[mas_sum[i]].minleng << " індекс/довжина шляху" << endl;
						v[mas_sum[i]].way = v[min].way;
						ss[t] << v[mas_sum[i]].number;
						ss[t] >> str;
						v[mas_sum[i]].way += ( " -> " + str);
						//cout << v[mas_sum[i]].way <<" шлях"<< endl;
						str = "";
						t++;
					}

				}
			}

		}
		//визначення яка з відстаней до суміжних вершин (нова чи стара) більша.


			v[min].proid = true;

		min = 29;
		m = 0;
		k = 0;
		x = 0;
		minleng = 1000;
		delete[] mas_v;
		delete[] mas_sum;

	}

	cout <<endl<<"Мінімальна відстань до даної вершини: " << v[end-1].minleng << endl;
	cout << "Шлях до неї: " << v[end-1].way;

	delete[] reb;
	return 0;
}
```

Вміст файлу:

6

1

2

1

2

3

1

3

4

3

4

5

3

5

6

8

7

8

1

8

9

1

9

10

2

10

11

7

11

12

7

13

14

1

14

15

2

15

16

3

16

17

7

17

18

7

19

20

3

20

21

8

21

22

1

22

23

5

23

24

4

25

26

7

26

27

3

27

28

3

28

29

6

29

30

4

1

7

5

7

13

5

13

19

8

19

25

2

2

8

1

8

14

1

14

20

2

20

26

3

3

9

7

9

15

7

15

21

1

21

27

4

4

10

4

10

16

4

16

22

3

22

28

5

5

11

1

11

17

2

17

23

3

23

29

4

6

12

3

12:

18

8

18

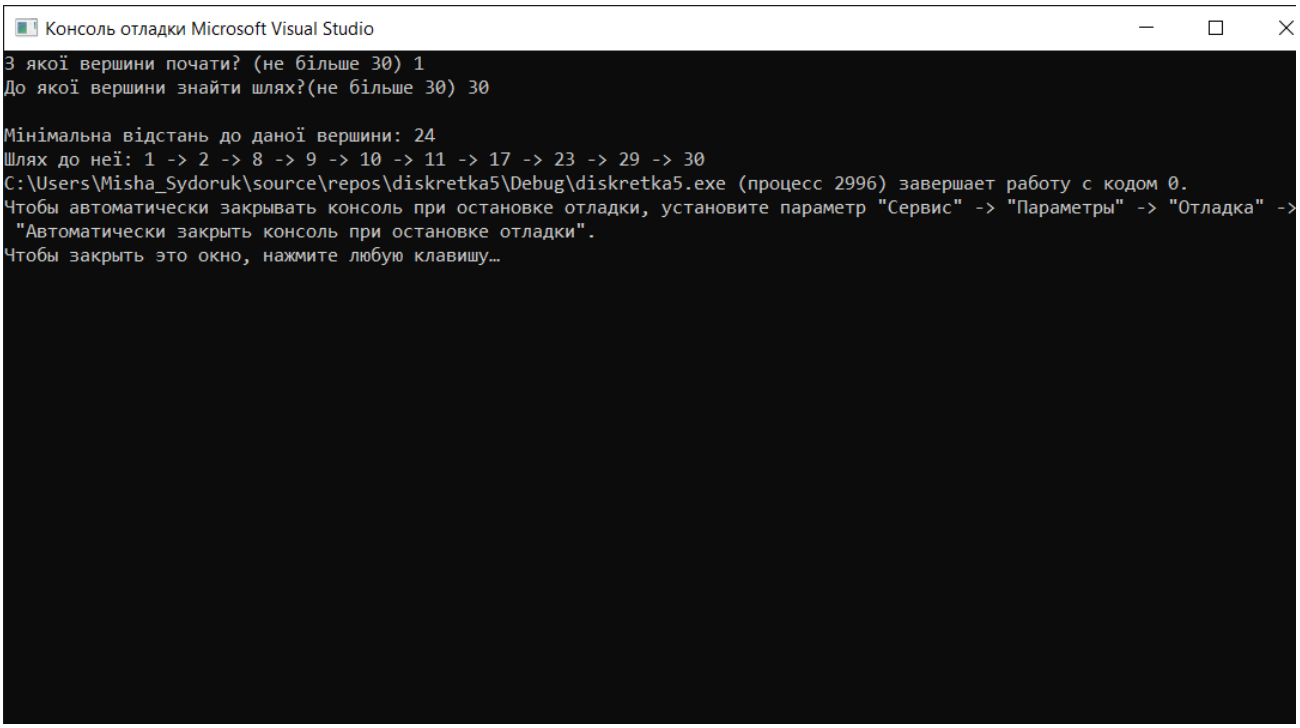
24

7

24

30

Результат виконання програми:



Висновок: : Отже, я набув практичних вмінь та навичок з використання алгоритму Дейкстри.