МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ «ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Програмна інженерія та інформаційні технології управління»

Звіт з лабораторної роботи №6

З предмету «Алгоритми та структури даних»

Виконав Студент групи КН-36а

Рубан Ю.Д.

Перевірила:

ас. Бородіна І. О.

Харків 2017

ФУНДАМЕНТАЛЬНІ АЛГОРИТМИ НА ГРАФАХ І ДЕРЕВАХ

Мета роботи: познайомитися зі способами представлення графів та отримати навички програмування алгоритмів, що їх обробляють.

Завдання

Розробити програму, яка читає з клавіатури числа N, M (1 < N, M < 256) — кількість вершин та ребер графу; послідовність M пар цілих чисел - ребра графу. Програма зберігає граф та виконує над ним алгоритм згідно варіанту.

Варіант завдання

Матриця суміжності.

Побудувати остовне дерево алгоритмом Прима.

Хід виконання роботи

Розроблено програму, яка читає з клавіатури числа N, M (1 < N, M < 256) — кількість вершин та ребер графу; послідовність M пар цілих чисел - ребра графу. Програма зберігає граф та виконує над ним алгоритм згідно варіанту.

Код програми:

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

class Graf
{
private:
	int** graf;
	int size;
public:
	Graf(int n)
	{
		graf = new int*[n];
		for (int i = 0; i < n; i++)
		{
			graf[i] = new int[n];
		}
		for (int i = 0; i < n; i++)
		{
			for (int j = 0; j < n; j++)
			{
				graf[i][j] = -1;
			}
		}
		size = n;

	}
	void add(int x, int y, int weight=1)
	{
		graf[x][y] = weight;
		graf[y][x] = weight;
	}
	void showTable()
	{
		for (int i = 0; i < size; i++)
		{
			for (int j = 0; j < size; j++)
			{
				cout << graf[i][j] << " ";
			}
```

```
                        cout << endl;
                }
        }
        Graf buildTree()
        {
                Graf temp = *this;
                Graf Tree(size);
                int x=0, y=0;
                int min = 10000;
                bool *markedX = new bool[size];
                for (int i = 0; i < size; i++)
                {
                        markedX[i] = false;
                }
                while (!marked(markedX, size))
                {
                        for (int i = 0; i < temp.size; i++)
                        {
                                temp.graf[i][y] = -1;
                        }
                        markedX[y] = true;
                        for (int i = size-1; i >= 0; i--)
                        {
                                for (int j = size - 1; j >=0; j--)
                                {//-----------------------
                                        if (markedX[i])
                                        {
                                                if (min >= temp.graf[i][j] && graf[i][j] != -1)
                                                {
                                                        min = temp.graf[i][j];
                                                        x = i;
                                                        y = j;
                                                }
                                        }
                                }//------------------------

                        }//////////////////////////
                        Tree.add(x, y, min);
                        min = 10000;
                }

                return Tree;
        }
        int getMinIndexY(int *a,int size,int Min)
        {
                for (int i = 0;i < size; i++)
                {
                        if (a[i] == Min) { return i; }
                }
                return -1;
        }
        bool marked(bool *m, int size)
        {
                int flag = 0;
                for (int i = 0; i < size; i++)
                {
                        if (m[i] == true)
                        {
                                flag++;
                        }
                }
                if (flag >= size - 1) { return true; }
                else
                        return false;
        }

};

int main()
{
        int N, M;
        cout << "enter number of graf`s tops" << endl;
        cin >> N;
        Graf g(N);
        cout << "enter number of arcs" << endl;
        cin >> M;
        int a, b,c;
```

```
	for (int i = 0; i < M; i++)
	{
		cout << "enter x, y (x->y)(y->x), and arc`s weight" << endl;
		cin >> a >> b>>c;
		g.add(a-1, b-1, c);
	}
	g.buildTree().showTable();

	system("pause");
	return 0;
}
```

Висновки

У даній лабораторній роботі я познайомився зі способами задання графа (матриця та таблиця суміжності), а також навчився опрацьовувати граф за допомогою різних алгоритмів.