Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

**МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ**

Кафедра програмування та захисту інформації

**Звіт**

з виконаної лабораторної роботи № 6

дисципліни “ Дискретна математика ”

на тему

«Алгоритми на скінчених графах»

Виконав :

студент академічної групи КІ-15

Аннаєв А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перевірив :

Викладач

Гермак В.С.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кропивницкий- 2017

**Лабораторная работа №6**

**ТЕМА:** Линейный синтез конечных графов

**ЦЕЛЬ:** Получить навыки линейного синтеза дискретная объектов и анализа наследования свойств.

Задание :

Выполнить построение остовного дерева графа из приложения 1 номер которого совпадает с номером вашей фамилии в списке группы, как вручную, так и с помощью программного средства методом в глубину графа;

1

2

5

3

4

6

5

1

3

1

2

5

4

10

7

1

2

5

3

4

6

5

1

1

2

4

/*\** *Алгоритм* Прима - *Краскала*

*\*\* Программирование = Лабараторная работа №6*

*\*\* Аннаев Арслан КИ-15*

*\** *\*/*

#include <iostream>

#include <cstdio>

#include <vector>

#include <algorithm>

*using* *namespace* std;

*struct* edge *//* *структура* *хранимая* *начальное* *и* *конечное* *положение* *и* *вес* *ребра*

{

int x, y, w;

edge(){}

edge(int x, int y, int w): x(x), y(y), w(w) {}

};

bool cmp(*const* edge& a, *const* edge& b) *//* *если* *вес* *ребра* *больше*

{

*return* a.w < b.w;

}

vector <int> leader; *//* *посещеный* *вектор*

int getLeader(int x) *//* *проверка* *посещали* *ли* *мы* *эту* *вершину*

{

*if* (x == leader[x])

*return* x;

*return* leader[x] = getLeader(leader[x]);

}

bool unite(int x, int y) *//* *обьединение* *двух* *вершин*

{

x = getLeader(x);

y = getLeader(y);

*if* (x == y)

*return* *false*;

*if* (rand() % 2 == 0)

swap(x, y);

leader[x] = y;

*return* *true*;

}

int main()

{

int n, m;

cout << "Enter Vertex count \_"; *//* *приглашение* *ввода* *кол.* *вершин*

cin >> n; *//* *ввод* *значения*

cout << "\nEnter count of edge ";*//* *приглашение* *ввода* *кол.* *ребер*

cin >> m; *//* *ввод* *значения*

*//* *массив* *структур* *хранящий* *начальную* *вершину* *и* *конечную* *с* *весом* *ребра*

vector <edge> e(m);

*//* *Определение* *графа*

*for* (int i = 0; i < m; i++)

{

cout << " Enter vertex \"x\" (" << i+1 << ") = "; *//* *приглашение*

cin>> e[i].x ; *//* *ввод* *значения*

cout << " Enter vertex \"y\" (" << i+1 << ") = "; *//* *приглашение*

cin >> e[i].y ; *//* *ввод* *значения*

cout << " Enter weigth \"w\" (" << i+1 << ") = "; ; *//* *приглашение*

cin>> e[i].w; *//* *ввод* *значения*

e[i].x--; *//* *декремент* *x*

e[i].y--; *//* *декремент* *y*

}

sort(e.begin(), e.end(), cmp); *//* *сортировка* *по* *возрастанию*

leader.resize(n); *//* *изменение* *размера*

*for* (int i = 0; i < n; i++)

leader[i] = i;

vector <edge> ans; 4149 6258 0506 0116

*for* (int i = 0; i < m; i++)

{

int x = e[i].x, y = e[i].y;

*if* (unite(x, y))

ans.push\_back(e[i]);

}

int sum = 0;

*for* (int i = 0; i < ans.size(); i++)

sum+=ans[i].w;

cout << "result weight = "<< sum;

*for* (int i = 0; i < ans.size(); i++)

cout << "( x = " << ans[i].x + 1 << "; y = " << ans[i].y + 1 << "), ";

*return* 0;

}

