ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| старший преподаватель |  |  |  |  |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ  «АЛГОРИТМЫ НА ГРАФАХ» |
|  |
| по дисциплине: СТРУКТУРЫ И АЛГОРИТМЫ ОБРАБОТКИ ДАННЫХ |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4131 |  |  |  | В.А. Алексеев |
|  |  |  | подпись, дата |  | инициалы, фамилия |

1. Цель работы

Целью работы является изучение графов и получение практических навыков их использования.

1. Задание на лабораторную работу

Разработать на языке программирования высокого уровня программу, которая должна выполнять поиск минимального реберного и минимального вершинного покрытия графа по заданной матрице инцедентности.

1. Листинг

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

bool found(vector<int>& a, int n, int sz) {

for (auto i = 0; i < sz; i++) {

if (a[i] == n)

return true;

}

return false;

}

void main() {

setlocale(LC\_ALL, "rus");

int n, r;

cout << "Введите колво вершин : ";

cin >> n;

cout << "Введите колво ребер : ";

cin >> r;

int\*\* a = new int\* [n];

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

a[i] = new int[r];

}

int\*\* b = new int\* [n];

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

b[i] = new int[r];

}

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

for (int j = 0; j < r; j++) {

cin >> a[i][j];

b[i][j] = a[i][j];

cout<< "\r";

}

}

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

for (int j = 0; j < r; j++) {

cout << a[i][j] << " ";

}

cout << endl;

}

//

vector<int > p;

int sum = 1;

while (sum != 0) {

int\* sumlin = new int[n]();

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

for (int j = 0; j < r; j++) {

sumlin[i] += abs(a[i][j]);

}

}

int maxlin = sumlin[0], index = 0;

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

if (sumlin[i] > maxlin) {

maxlin = sumlin[i];

index = i;

}

}

sum = 0;

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

sum += sumlin[i];

}

if (sum != 0) {

p.push\_back(index);

for (int j = 0; j < r; j++) {

if (a[index][j] != 0) {

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

a[i][j] = 0;

}

}

}

}

}

cout << "Минимальное вершинное покрытие : ";

for (int s : p) {

cout << s + 1 << " ";

}

cout << endl;

struct firstdel {

int v;

int r;

};

vector<firstdel> min;

vector<int> reb;

vector<int> ver;

//поиск вершин с одной дугой и запись в минреб покрытие

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

int s = 0, ind = 0;

for (int j = 0; j < r; j++) {

if (b[i][j] == 1) {

s += 1;

ind = j;

}

}

if (s == 1) {

firstdel v2;

reb.push\_back(ind);

for (int k = 0; k < n; k++) {

if (b[k][ind] == 1) {

ver.push\_back(k);

v2.r = ind;

v2.v = k;

min.push\_back(v2);

}

}

}

}

ver.erase(unique(ver.begin(), ver.end()), ver.end());

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

b[min[i].v][min[i].r] = 0;

//поиск вершин которых нет в минреб

if (ver.size() != n) {

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

int s = 0;

int v[] = { -1,-1 };

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

if (b[j][i] == 1) {

v[s] = j;

s += 1;

}

}

if (found(ver, v[0], ver.size()) == 0 && found(ver, v[1], ver.size()) == 0 && v[0] != -1) {

ver.push\_back(v[0]);

ver.push\_back(v[1]);

reb.push\_back(i);

b[v[0]][i] = 0;

b[v[1]][i] = 0;

}

}

}

ver.erase(unique(ver.begin(), ver.end()), ver.end());

if (ver.size() != n) {

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

int s = 0;

int v[] = { -1,-1 };

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

if (b[j][i] == 1) {

v[s] = j;

s += 1;

}

}

if ((find(ver.begin(), ver.end(), v[0]) == ver.end()) && (find(ver.begin(), ver.end(), v[1]) != ver.end()) || (find(ver.begin(), ver.end(), v[0]) != ver.end()) && (find(ver.begin(), ver.end(), v[1]) == ver.end())) {

ver.push\_back(v[0]);

ver.push\_back(v[1]);

reb.push\_back(i);

b[v[0]][i] = 0;

b[v[1]][i] = 0;

}

}

}

cout << "Минимальное реберное покрытие : ";

for (int s : reb)

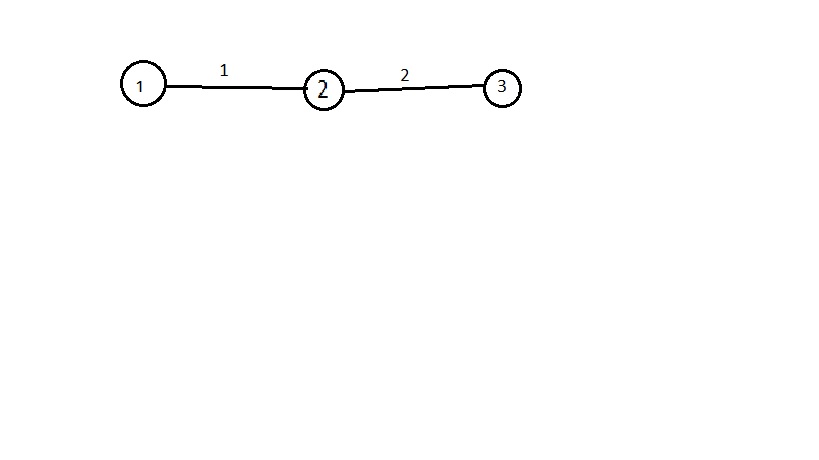
cout << " " << s + 1;

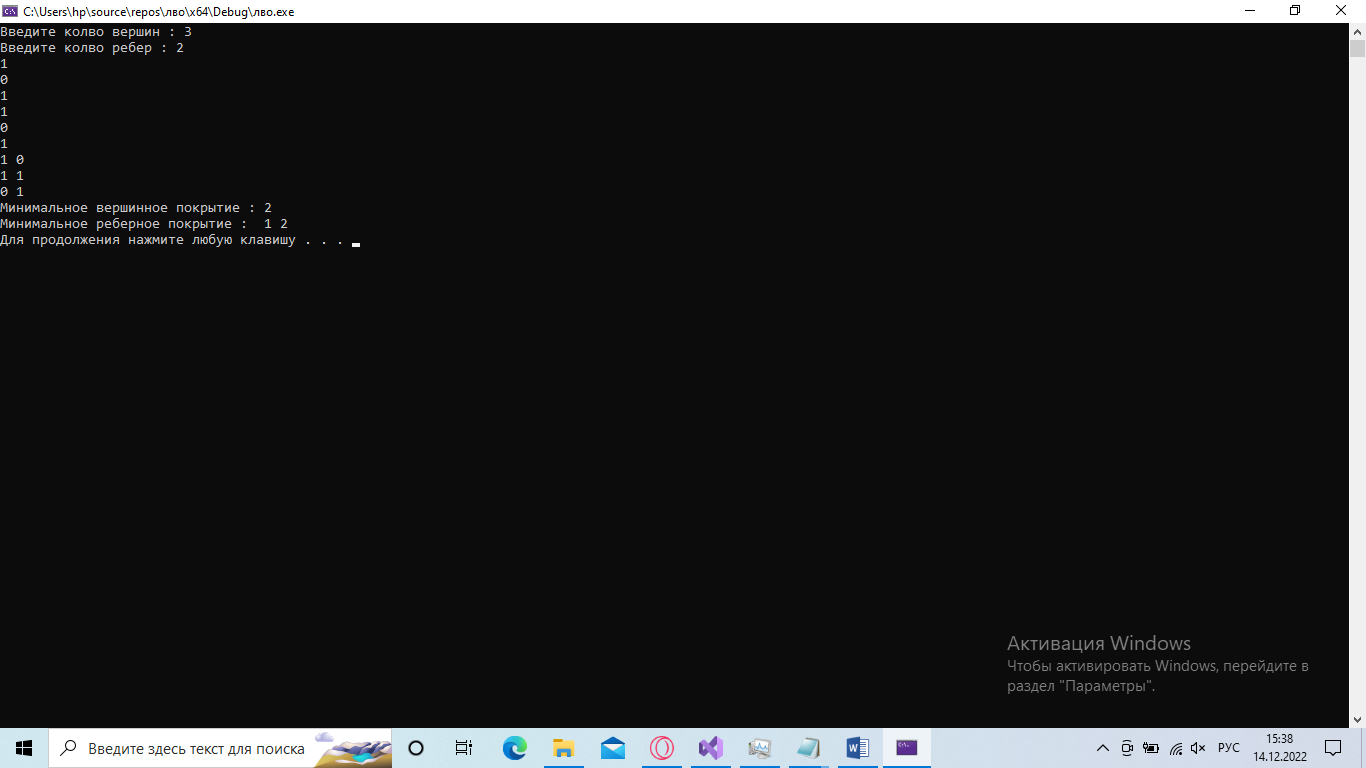
cout << endl;

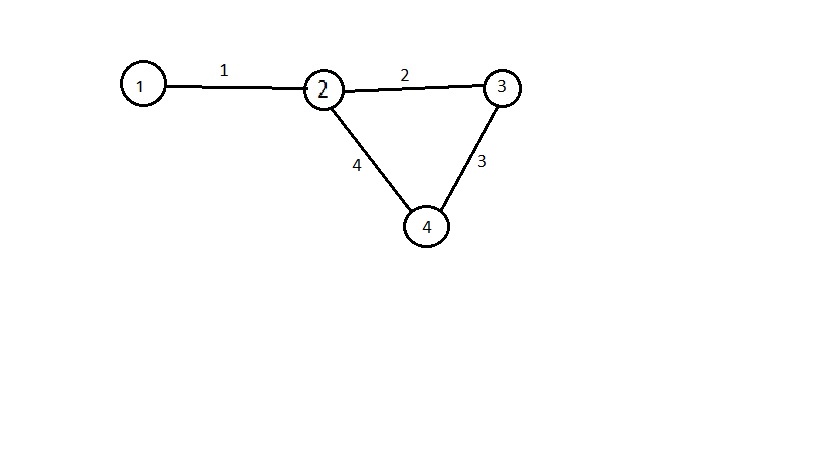
system("pause");

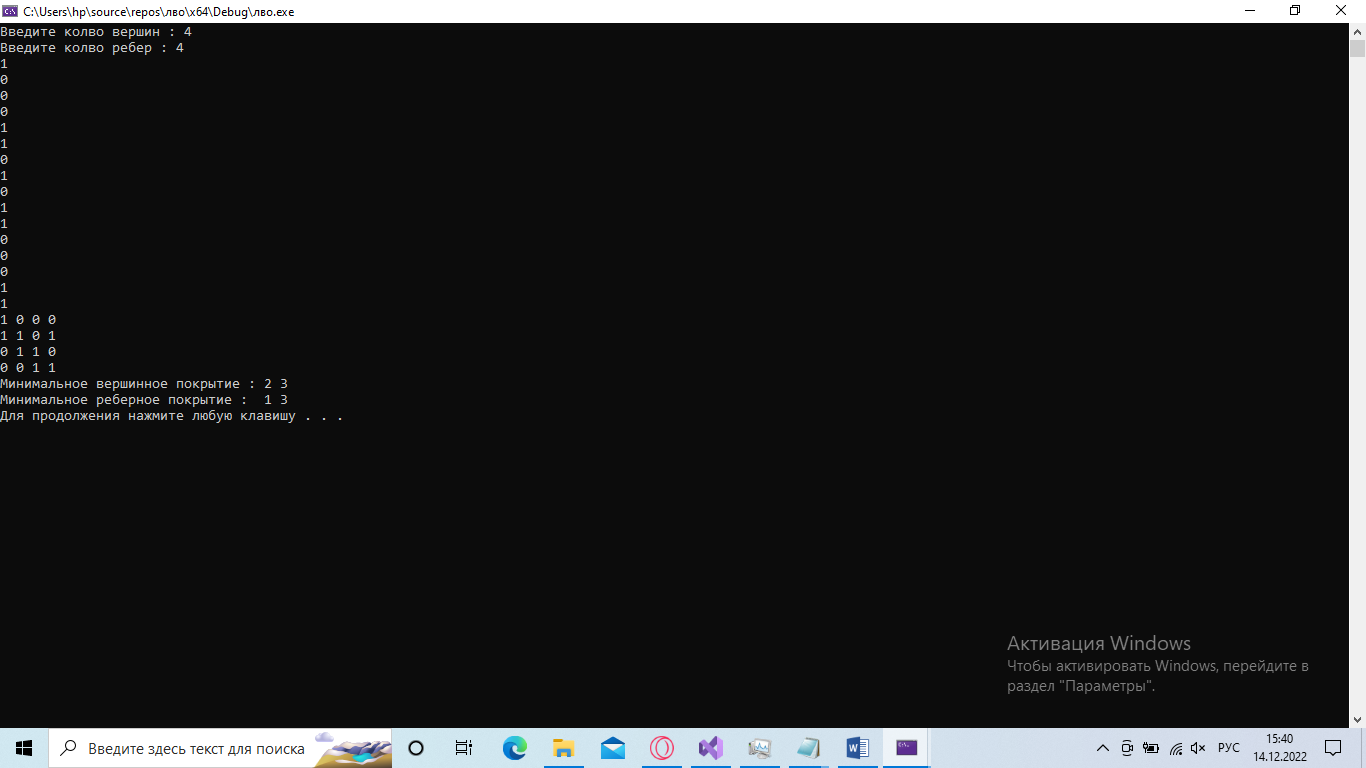
}

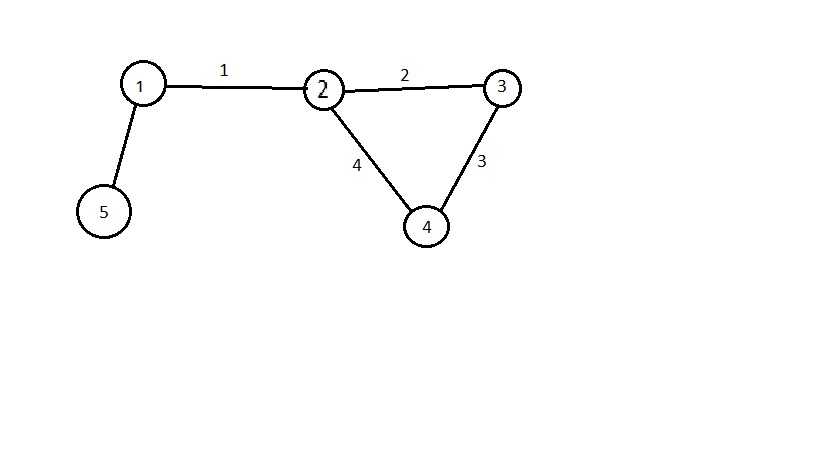
* 1. Работа программы

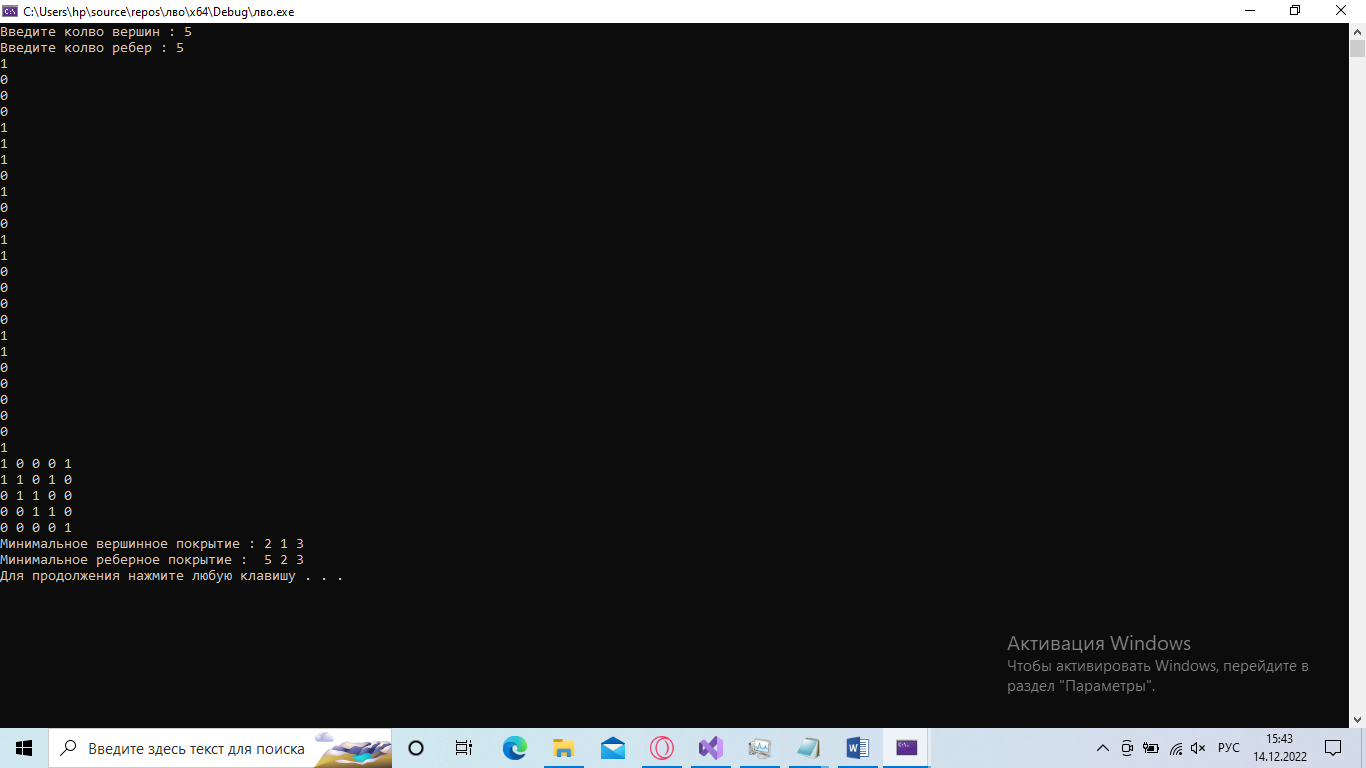












* 1. Вывод

Я изучил графы и получил практические навыки их использования и программирования.