МИНИСТЕРВСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра «Вычислительная техника»

ОТЧЕТ

по лабораторной работе №5

по дисциплине: "Логика и основы алгоритмизации в инженерных задачах"

на тему: "Определение характеристик графов"

Выполнили студенты группы 24ВВВ3:

Пяткин Р. С.

Гусаров Е. Е.

Принял:

к.т.н., доцент, Юрова О. В.

к.т.н., Деев М. В.

Пенза 2025

Цель

Изучение характеристик графов.

Лабораторное задание

Задание 1

Сгенерируйте (используя генератор случайных чисел) матрицу смежности для неориентированного графа G. Выведите матрицу на экран.

Определите размер графа G, используя матрицу смежности графа.

Найдите изолированные, концевые и доминирующие вершины.

Задание 2

Постройте для графа G матрицу инцидентности.

Определите размер графа G, используя матрицу инцидентности графа.

Найдите изолированные, концевые и доминирующие вершины.

Результаты программ

```
ruslan@DexpAtlas:~/Документы/Algorithmization/LAB_5$ ./test
Введите кол-во вершин в матрице: 4
=====РЕЗУЛЬТАТ=====
0) 0   0   0   1
1) 0   0   0   0
2) 0   0   1   1
3) 1   0   1   1
Концевая вершина: 0
Изолированная вершина: 1
Доминирующая вершина:  2
Размер графа: 4
ruslan@DexpAtlas:~/Документы/Algorithmization/LAB_5$
```

Рис. 1 — lab_5

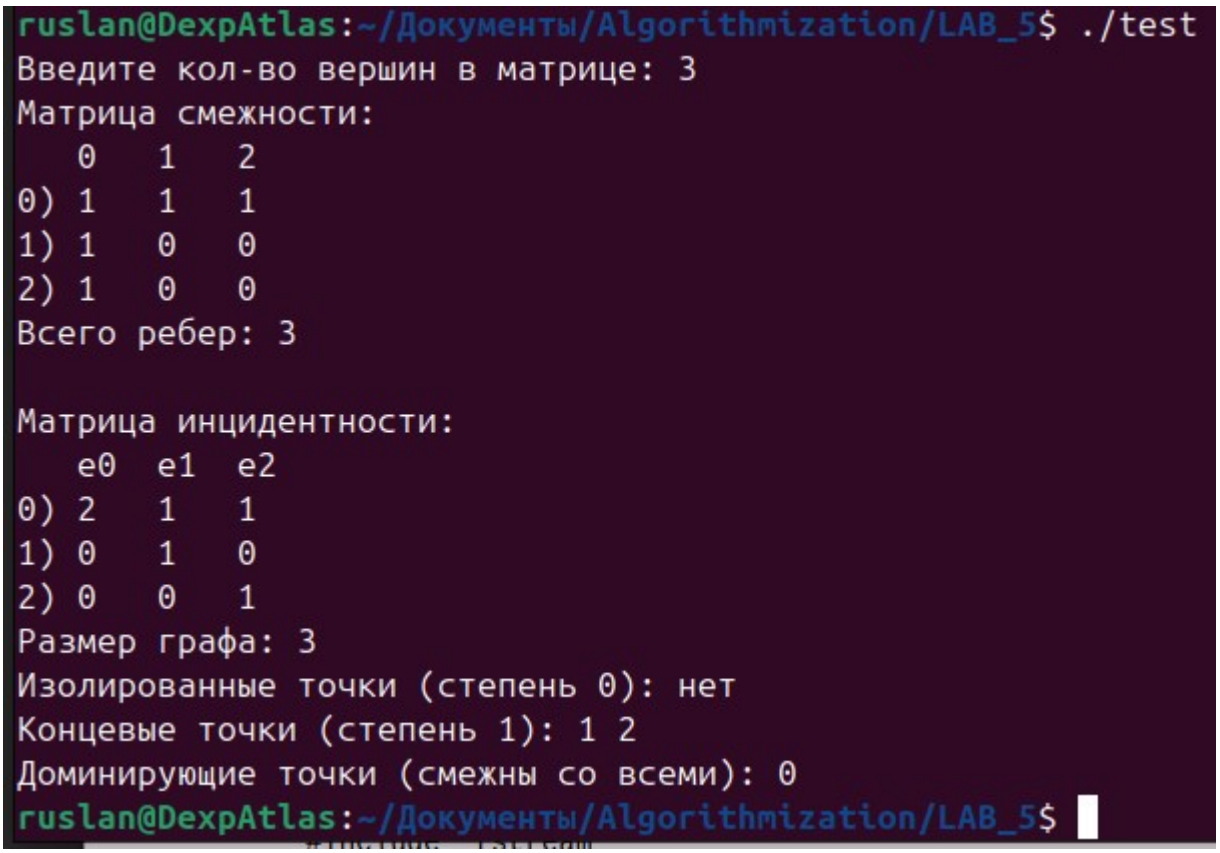

Рис. 2 — lab_5_2

Вывод:

В ходе выполнения лабораторной работы были разработаны программы для выполнения заданий Лабораторной работы №5. В процессе выполнения работы были использованы знания о характеристиках графов.

Листинг

Файл lab_5.cpp

```
#include <fstream>

#include <iostream>
#include <ctime>

using namespace std;

int main(){
srand(time(0));
int N, Size = 0;
int** G = new int*[N];
for (int i = 0; i < N; i++) {
G[i] = new int[N];
}
int* loop = new int[N];
int* deg = new int[N];
```

```
cout<<"Введите кол-во вершин в матрице: ";
cin>>N;
for(int i = 0; i < N; i++){
for(int j = i; j < N; j++){
G[j][i] = G[i][j] = rand() % 2;
Size += G[i][j];
}
}

cout<<"=====РЕЗУЛЬТАТ====="<<endl;
for(int i = 0; i < N; i++){
cout<<i<<") ";
for(int j = 0; j < N; j++){
cout<<G[i][j]<<" ";
}
cout<<endl;
}

for(int i = 0; i < N; i++){
for(int j = 0; j < N; j++){
if(i == j){
loop[i] = G[i][j];

}else{
deg[i] += G[i][j];
}
deg[i] += loop[i] * 2;
}
}

for(int i = 0; i < N; i++){
if(loop[i] == 0 && deg[i] == 0 || loop[i] == 1 && deg[i] == 2){
cout<<"Изолированная вершина: "<<i<<endl;
}
if(deg[i] == 1){
cout<<"Концевая вершина: "<<i<<endl;
}
if((deg[i] - 2*loop[i]) == (N - 1)){
cout<<"Доминирующая вершина: "<<i<<endl;
}
}
cout<<"Размер графа: "<<Size<<endl;

for (int i = 0; i < N; i++) {
delete[] G[i];
}
delete[] loop;
delete[] deg;
}
```

Файл lab_5_2.cpp

```
#include <fstream>

#include <iostream>
#include <ctime>
#include <vector>

using namespace std;

int main() {
srand(time(0));
int N = 0;
int Size = 0;
cout << "Введите кол-во вершин в матрице: ";
cin >> N;
int* loop = new int[N];
int* deg = new int[N];

int** G = new int*[N];
for (int i = 0; i < N; i++) {
G[i] = new int[N]();
}

for (int i = 0; i < N; i++) {
for (int j = i; j < N; j++) {
G[i][j] = G[j][i] = rand() % 2;
if (G[i][j] == 1) {
Size++;
}
}
}

cout << "Матрица смежности:" << endl;
cout << " ";
for (int i = 0; i < N; i++) {
cout << i << " ";
}
cout << endl;
for (int i = 0; i < N; i++) {
cout << i << ") ";
for (int j = 0; j < N; j++) {
cout << G[i][j] << " ";
}
cout << endl;
}

cout << "Всего ребер: " << Size << endl;

int** H = nullptr;
```

```
if (Size > 0) {
H = new int*[N];
for (int i = 0; i < N; i++) {
H[i] = new int[Size]();
}
int currentEdge = 0;
for (int i = 0; i < N; i++) {
for (int j = i; j < N; j++) {
if (G[i][j] == 1) {
if (i == j) {
H[i][currentEdge] = 2;
} else {
H[i][currentEdge] = 1;
H[j][currentEdge] = 1;
}
currentEdge++;
}
}
}

cout << "\nМатрица инцидентности:" << endl << " ";
for (int j = 0; j < Size; j++) {
cout << "e" << j << " ";
}
cout << endl;
for (int i = 0; i < N; i++) {
cout << i << ") ";
for (int j = 0; j < Size; j++) {
if (H[i][j] == 2)
cout << H[i][j] << " ";
else
cout << H[i][j] << " ";
}
cout << endl;
}

vector<int> isolated;
vector<int> pendant;
vector<int> dominant;
int* degrees = new int[N]();
for (int i = 0; i < N; i++) {
for (int j = 0; j < Size; j++) {
if (H[i][j] == 1) {
degrees[i]++;
} else if (H[i][j] == 2) {
degrees[i] += 2;
}
}
if (degrees[i] == 0) {
isolated.push_back(i);
```

```
} else if (degrees[i] == 1) {
pendant.push_back(i);
}
}
for (int i = 0; i < N; i++) {
bool isDominant = true;
for (int j = 0; j < N; j++) {
if (i != j) {
bool connected = false;
for (int k = 0; k < Size; k++) {
if ((H[i][k] == 1 && H[j][k] == 1) ||
(H[i][k] == 2 && i == j)) {
connected = true;
break;
}
}
if (!connected) {
isDominant = false;
break;
}
}
}
if (isDominant) {
dominant.push_back(i);
}
}
cout << "Размер графа: " << Size << endl;
cout << "Изолированные точки (степень 0): ";
if (isolated.empty()) {
cout << "нет";
} else {
for (int i = 0; i < isolated.size(); i++) {
cout << isolated[i] << " ";
}
}
cout << endl;
cout << "Концевые точки (степень 1): ";
if (pendant.empty()) {
cout << "нет";
} else {
for (int i = 0; i < pendant.size(); i++) {
cout << pendant[i] << " ";
}
}
cout << endl;
cout << "Доминирующие точки (смежны со всеми): ";
if (dominant.empty()) {
cout << "нет";
} else {
for (int i = 0; i < dominant.size(); i++) {
```

```
cout << dominant[i] << " ";
}
}
cout << endl;
delete[] degrees;
} else {
cout << "В графе нет ребер!" << endl;
}

for (int i = 0; i < N; i++) {
delete[] G[i];
}
delete[] G;
if (Size > 0) {
for (int i = 0; i < N; i++) {
delete[] H[i];
}
delete[] H;
}

return 0;
}
```