Пензенский государственный университет

Кафедра ”Вычислительной техники ”

**Отчет**

по лабораторной работе №7

## по дисциплине: “ Унарные и бинарные операции над графами”

## по теме: “ Определение характеристик графов”

## *Выполнил студент группы 19ВВ1:*

## Васильев Л.Р

## *Приняли:*

## Митрохин М.А

ПЕНЗА 2020

Цель: Научится определять характеристики графов

Общие сведения.

Если G - граф, содержащий непустое множество n вершин V и

множество ребер E и d(v i , v j ) – расстояние между двумя произвольными

вершинами v i  и v j , тогда для фиксированной вершины v величина

),(max)(jvvdve

,

где v, v j V и j = 1…n называется эксцентриситетом вершины v i .

Другими словами эксцентриситет вершины – расстояние до наиболее

удаленной вершины графа.

Например, для вершины 1 графа, показанного на рисунке 1

эксцентриситет равен 7.

Рисунок 1 – Граф

Максимальный эксцентриситет среди эксцентриситетов всех вершин

графа называется диаметром графа G и обозначается через D(G).

Вершина v i называется периферийной, если её эксцентриситет равен

диаметру графа e(v i ) = d(G).

Минимальный из эксцентриситетов вершин графа называется его

радиусом и обозначается через r(G).

Вершина v i называется центральной, если её эксцентриситет равен

радиусу графа e(v i ) = r(G).

Множество всех центральных вершин графа называется его центром.

Граф G может иметь единственную центральную вершину или несколько

центральных вершин.

Степенью вершины графа G называется число инцидентных ей ребер.

Степень вершины v i обозначается через deg(v i ).

Вершина v i со степенью 0 называется изолированной, со степенью 1 –

концевой.

Вершина графа, смежная с каждой другой его вершиной, называется

доминирующей.

Листинг:

void BFSD(int p,int arr, int max, int min);

int \*\*M1, \*visit;

/\*

if(max<visit[p] && visit[p] != 1000){

max = visit[p];

}else if(min>visit[p] && visit[p] != 0){

min = visit[p];

}

\*/

void BFSD(int p, int arr, int max, int min){

int k=0;

std::queue <int> Q;

Q.push(p);

visit[p] = 0;

printf("\n");

while(!Q.empty()){

p = Q.front();

Q.pop();

k+=1;

printf("v[%d]=%d ",k, p);

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

if(M1[p][i] > 0 && visit[i] > visit[p] + M1[p][i]){

Q.push(i);

visit[i] = visit[p] + M1[p][i];

}

}

}

}

int main()

{

srand(time(NULL));

int max = 0, min=100000000;

int p;

int arr, q=0;

int \*s, \*b, \*m, \*n, \*l;

float k;

printf("Введем матрицу");

scanf("%d",&arr );

M1 = (int\*\*)malloc(arr\*sizeof(int));

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

M1[i] = (int \*)malloc(arr\*sizeof(int));

}

visit = (int\*)malloc(arr\*sizeof(int));

s = (int \*)malloc(arr\*sizeof(int));

b = (int \*)malloc(arr\*sizeof(int));

m = (int \*)malloc(arr\*sizeof(int));

n = (int \*)malloc(arr\*sizeof(int));

l = (int \*)malloc(arr\*sizeof(int));

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

visit[i] = 1000;

}

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

s[i] = 0;

}

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

b[i] = 0;

}

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

m[i] = 0;

}

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

n[i] = 0;

}

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

l[i] = 0;

}

printf("Введем матрицу:\n");

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

for(int j = 0;j<i+1;j++){

k = rand()%10;

if(k>5){

k = 0;

}

M1[i][j] = k;

M1[j][i] = k;

}

}

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

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

printf("%d ",M1[i][j] );

}

printf("\n");

}

printf("\n");

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

p = i;

printf("\n");

BFSD(p, arr, max, min);

printf("\n");

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

printf("\n%d", visit[i]);

if(max<visit[i] && visit[i] != 1000){

max = visit[i];

}else if(min>visit[i] && visit[i] != 0){

min = visit[i];

if(min==1000){

min = max;

}

}

}

printf("\n Выведем переферийные вершины:\n");

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

if(visit[i] == max){

s[i] = i;

printf("%d ", i);

s[i] = 0;

}

}

printf("\n Выведем центральные вершины:\n");

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

if(visit[i] == min){

b[i] = i;

printf("%d ", i);

b[i] = 0;

}

}

printf("\nИзолированная вершина ");

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

if(visit[i] == 1000){

m[i] = i;

printf("%d ", i);

m[i] = 0;

}

}

printf("\nКонцевой вершина ");

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

for(int j = 0; j< i+1; j++){

if(i == j && M1[i][j] != 0){

n[i] = i;

printf("%d ", i);

n[i] = 0;

}

}

}

printf("\nДоминирующая вершина ");

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

for(int j=0;j<i+1;j++){

if(M1[i][j] != 0){

q++;

if(q == arr){

l[i] = i;

printf("%d", l[i]);

}

}

}

}

printf("\nДиаметр = %d, радиус = %d",max, min);

max = 0;

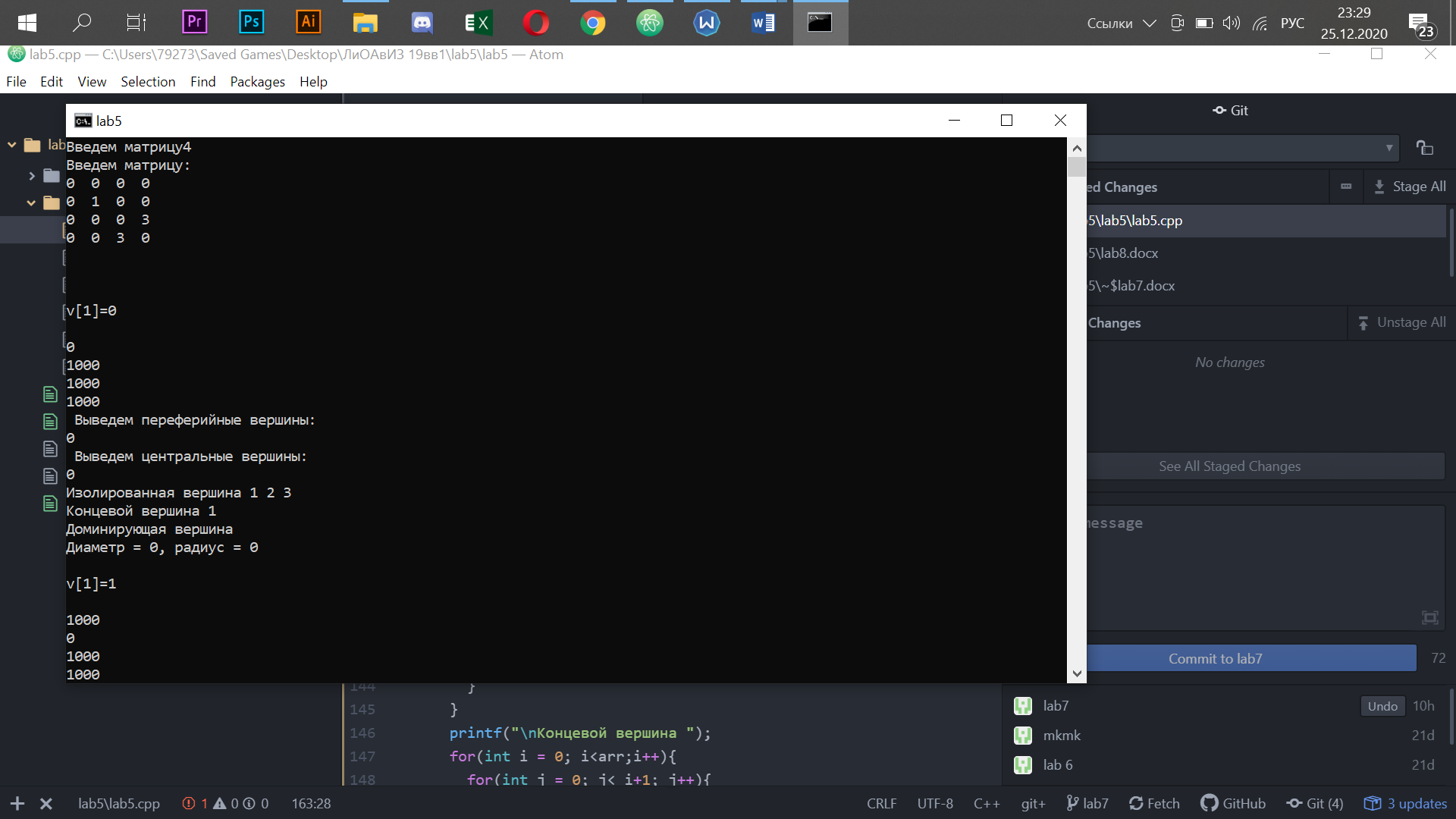
min = 1000000000;

visit[i] = 1000;

}

}

Работа программы



Вывод: Мы научились определять характеристики графов