Пензенский государственный университет

Кафедра ”Вычислительной техники ”

**Отчет**

по лабораторной работе №7

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

## по теме: “ Поиск расстояний во взвешенном графе”

## *Выполнил студент группы 19ВВ1:*

## Васильев Л.Р

## *Приняли:*

## Митрохин М.А

ПЕНЗА 2020

Цель: Изучить поиск расстояний во взвешенном графе.

Во взвешенном графе в отличие от не взвешенного каждое ребро имеет

вес, отличный от нуля. Поэтому в матрице смежности взвешенного графа

содержится информация не только о наличии ребра, но и о его весе.

Рисунок 1 – Граф

Поиск расстояний между вершинами в таком графе также возможно

построить используя процедуры обхода графа. Отличие от поиска расстояний

в не взвешенном графе будет состоять в том, что при обновлении расстояния

до вершины при ее посещении оно будет увеличиваться не на 1, а на

величину веса ребра.

Листинг

void BFSD(int p,int arr);

int \*\*M1, \*visit;

void BFSD(int p, int arr){

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 p;

int arr;

int s = 0;

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));

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

visit[i] = 1000;

}

printf("Введем матрицу:\n");

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

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

k = rand()%10;

if(i == j){

k = 0;

}

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);

printf("\n");

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

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

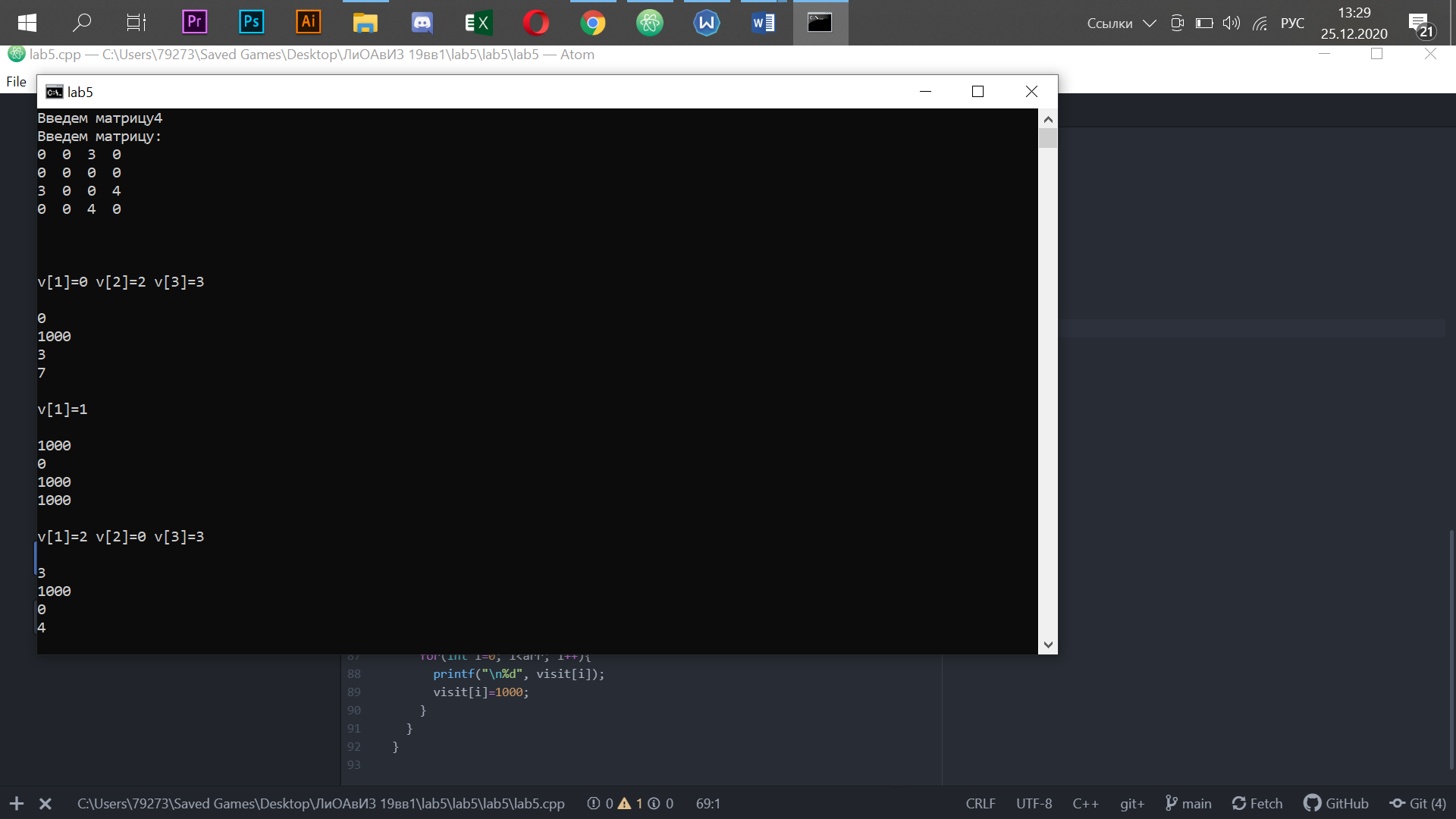
visit[i]=1000;

}

}

}

Схема работы программы



Вывод: мы изучили поиск расстояний во взвешенном графе.