Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

ОТЧЁТ

по лабораторной работе №5

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Унарные и бинарные операции над графами»

Выполнил:

студент группы 21ВВ2

Шилов Алексей

Приняли:

д.т.н. Митрохин М.А.

к.т.н. Юрова О.В.

Пенза 2022

Задание 1

Сгенерировать (используя генератор случайных чисел) две матрицы M1, М2 смежности неориентированных помеченных графов G1, G2. Вывести сгенерированные матрицы на экран.

Задание 2

Для матричной формы представления графов выполнить операцию:

а) отождествления вершин

б) стягивания ребра

в) расщепления вершины

Номера выбираемых для выполнения операции вершин ввести с клавиатуры.

Результат выполнения операции вывести на экран.

Задание 3

Для матричной формы представления графов выполнить операцию:

а) объединения

б) пересечения

в) кольцевой суммы

Результат выполнения операции вывести на экран.

Листинг

#include <iostream>

#include <time.h>

#include <stdlib.h>

using namespace std;

void Vivod(int\*\* M, int x){

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

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

cout << M[i][j] << " ";}

cout << "\n";}}

int\* Okr(int n, int m, int x, int \*Q1, int \*Q2, int \*Q3, int \*\*M){

for (int i = 0; i < x; i++) //Окружение выбранных вершин

{

if (M[n][i] == 1){

Q1[i] = 1;}

else

Q1[i] = 0;

if (M[m][i] == 1){

Q2[i] = 1;}

else

Q2[i] = 0;

if (Q1[i] == 1 or Q2[i] == 1) //Объединение окружений

{

Q3[i] = 1;}

else

Q3[i] = 0;}

return Q3; }

int\*\* S(int n, int m, int x, int y, int\*\* M) {

int\* Q1 = (int\*)malloc(x \* sizeof(int));

int\* Q2 = (int\*)malloc(x \* sizeof(int));

int\* Q3 = (int\*)malloc(x \* sizeof(int));

Okr(n, m, x, Q1, Q2, Q3, M);

M[n] = Q3;

for (int i = 0; i < x; i++)

M[i][n] = Q3[i];

if (n > m) {

int t = n;

n = m;

m = t; }

for (int i = n; i < x - 1; i++){

int\* t = (int\*)malloc(x \* sizeof(int));

t = M[i];

M[i] = M[i + 1];

M[i + 1] = t; }

for (int i = 0; i < x; i++)

for (int j = n; j < x - 1; j++){

int t = M[i][j];

M[i][j] = M[i][j + 1];

M[i][j + 1] = t; }

for (int i = m - 1; i < x - 1; i++){

int\* t = (int\*)malloc(x \* sizeof(int));

t = M[i];

M[i] = M[i + 1];

M[i + 1] = t; }

for (int i = 0; i < x; i++)

for (int j = m - 1; j < x - 1; j++){

int t = M[i][j];

M[i][j] = M[i][j + 1];

M[i][j + 1] = t; }

return M; }

int\*\* R(int n, int x, int y, int\*\* M) {

int\* Q1 = (int\*)malloc(x \* sizeof(int));

for (int i = 0; i < x; i++) //Окружение выбранной вершины

{

if (M[n][i] == 1) {

Q1[i] = 1; }

else

Q1[i] = 0; }

M[n] = Q1;

M[y - 1] = Q1;

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

M[i][n] = Q1[i];

M[i][y-1] = Q1[i]; }

for (int i = n; i < x - 1; i++){

int\* t = (int\*)malloc(x \* sizeof(int));

t = M[i];

M[i] = M[i + 1];

M[i + 1] = t; }

for (int i = 0; i < x; i++)

for (int j = n; j < x - 1; j++) {

int t = M[i][j];

M[i][j] = M[i][j + 1];

M[i][j + 1] = t; }

return M; }

int main(){

srand(time(NULL));

int x = 6, n = 0, m = 0, y = 6;

int z2 = 1000;

cout << "M1:" << "\n";

int\*\* M1 = (int\*\*)malloc(x \* sizeof(int)); //M1

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

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

for (int i = 0; i < x; i++) //Заполнение

{

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

M1[i][j] = rand() % 2;

M1[j][i] = M1[i][j]; }}

Vivod(M1, x);

cout << "\n" << "M2:" << "\n";

int\*\* M2 = (int\*\*)malloc(x \* sizeof(int)); //M2

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

M2[i] = (int\*)malloc(x \* sizeof(int)); }

for (int i = 0; i < x; i++) //Заполнение

{

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

M2[i][j] = rand() % 2;

M2[j][i] = M2[i][j]; }}

Vivod(M2, x);

//2 Задание

do{

cout << "\n" << "Otozdestvlenie: 1, Ctyagivanie: 2, Rascheplenie 3: ";

cin >> z2;

//Отождествление

if (z2 == 1) {

cout << "\n" << "Vvedite nomera 2 vershin:" << "\n";

cin >> n >> m;

n -= 1; //Для более удобной работы с массивами

m -= 1; //Для более удобной работы с массивами

y -= 1;

//Отождествление в массиве M1

S(n, m, x, y, M1);

M1 = (int\*\*)realloc(M1, y \* sizeof(int)); //M1

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

M1[i] = (int\*)realloc(M1[i], y \* sizeof(int)); }

cout << "M1:" << "\n";

Vivod(M1, y);

//Отождествление в массиве M2

S(n, m, x, y, M2);

M2 = (int\*\*)realloc(M2, y \* sizeof(int)); //M2

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

M2[i] = (int\*)realloc(M2[i], y \* sizeof(int)); }

cout << "M2:" << "\n";

Vivod(M2, y);

x = y; }

//Стягивание ребра

if (z2 == 2) {

do {

cout << "\n" << "Vvedite nomera 2 smeznih vershin v M1:" << "\n";

cin >> n >> m;

n -= 1; //Для более удобной работы с массивами

m -= 1; //Для более удобной работы с массивами

z2 = M1[n][m];

} while (z2 != 1);

y -= 1;

//Стягивание в массиве M1

S(n, m, x, y, M1);

M1 = (int\*\*)realloc(M1, y \* sizeof(int)); //M1

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

M1[i] = (int\*)realloc(M1[i], y \* sizeof(int)); }

cout << "M1:" << "\n";

Vivod(M1, y);

do {

cout << "\n" << "Vvedite nomera 2 smeznih vershin v M2:" << "\n";

cin >> n >> m;

n -= 1; //Для более удобной работы с массивами

m -= 1; //Для более удобной работы с массивами

z2 = M2[n][m];

} while (z2 != 1);

//Стягивание в массиве M2

S(n, m, x, y, M2);

M2 = (int\*\*)realloc(M2, y \* sizeof(int)); //M2

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

M2[i] = (int\*)realloc(M2[i], y \* sizeof(int)); }

cout << "M2:" << "\n";

Vivod(M2, y);

x = y;

z2 = 2; }

//Расщепление вершины

if (z2 == 3) {

cout << "\n" << "Vvedite nomer vershini" << "\n";

cin >> n;

n -= 1; //Для более удобной работы с массивами

y += 1;

M1 = (int\*\*)realloc(M1, y \* sizeof(int));

M2 = (int\*\*)realloc(M2, y \* sizeof(int));

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

M1[i] = (int\*)realloc(M1[i], y \* sizeof(int));

M2[i] = (int\*)realloc(M2[i], y \* sizeof(int)); }

M1[y - 1] = (int\*)malloc(y \* sizeof(int));

M2[y - 1] = (int\*)malloc(y \* sizeof(int));

//Расщепление В M1

R(n, x, y, M1);

cout << "M1:" << "\n";

Vivod(M1, y);

//Расщепление В M2

R(n, x, y, M2);

cout << "M2:" << "\n";

Vivod(M2, y);

x = y; }

} while (z2 != 0);

//Задание 3

int\*\* Mo = (int\*\*)malloc(x \* sizeof(int));

int\*\* Mp = (int\*\*)malloc(x \* sizeof(int));

int\*\* Mk = (int\*\*)malloc(x \* sizeof(int));

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

Mo[i] = (int\*)malloc(x \* sizeof(int));

Mp[i] = (int\*)malloc(x \* sizeof(int));

Mk[i] = (int\*)malloc(x \* sizeof(int)); }

//Объединение

for (int i = 0; i < x; i++)

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

if (M1[i][j] == 1 || M2[i][j] == 1)

Mo[i][j] = 1;

else

Mo[i][j] = 0; }

cout << "Ob'edinenie M1 i M2" << "\n";

Vivod(Mo, y);

//Пересечение

for (int i = 0; i < x; i++)

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

if (M1[i][j] == 1 && M2[i][j] == 1)

Mp[i][j] = 1;

else

Mp[i][j] = 0; }

cout << "Peresevhenie M1 i M2" << "\n";

Vivod(Mp, y);

//Кольцевая сумма

int\* zn = (int\*)malloc(x \* sizeof(int));

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

zn[i] = 0;

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

if (M1[i][j] != M2[i][j]) {

Mk[i][j] = 1; }

else {

Mk[i][j] = 0;

zn[i] += 1; } } }

int size = x;

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

if (zn[k] == x) {

cout << endl;

cout << endl;

Vivod(Mk, x);

cout << endl;

size -= 1;

for (int i = k; i < x - 1; i++) {

int\* t = (int\*)malloc(x \* sizeof(int));

t = Mk[i];

Mk[i] = Mk[i + 1];

Mk[i + 1] = t; }

for (int i = 0; i < x; i++)

for (int j = k; j < x - 1; j++) {

int t = Mk[i][j];

Mk[i][j] = Mk[i][j + 1];

Mk[i][j + 1] = t; } } }

Mk = (int\*\*)realloc(Mk, size \* sizeof(int)); //M1

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

Mk[i] = (int\*)realloc(Mk[i], size \* sizeof(int)); }

cout << "Kol'cevaya summa M1 i M2" << "\n";

Vivod(Mk, size);

free(M1);

free(M2);

free(Mo);

free(Mp);

free(Mk);

}

Результаты

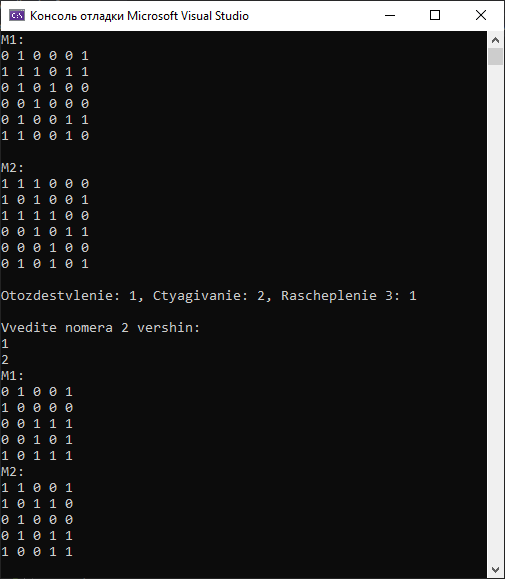


Рисунок 1 – Отождествление вершин

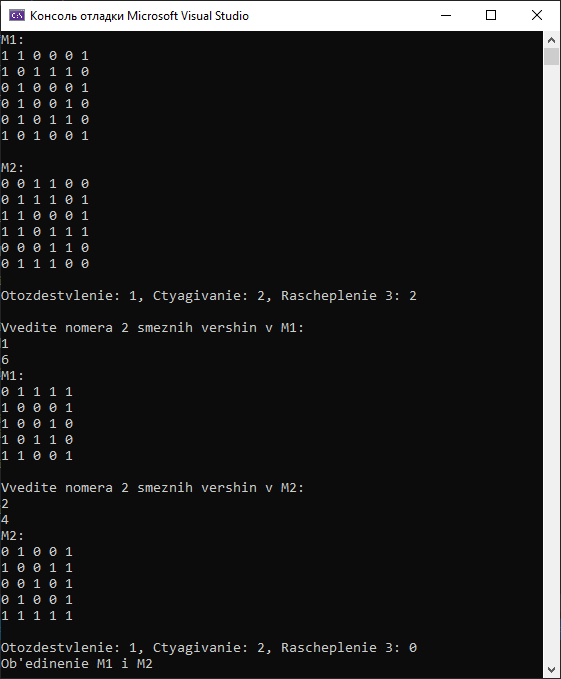


Рисунок 2 – Стягивание смежных вершин

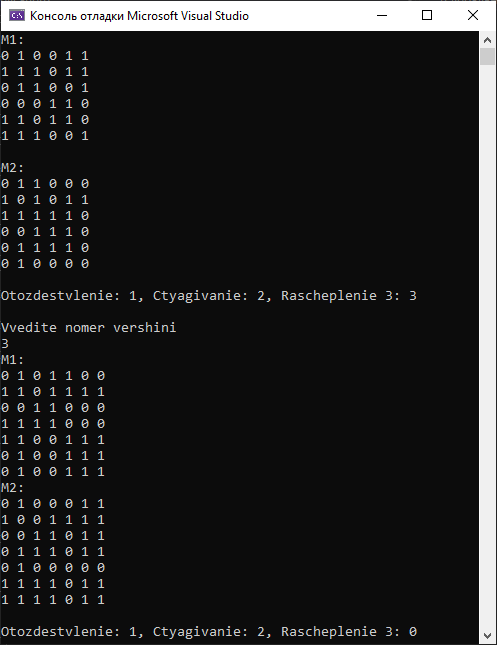


Рисунок 3 – Расщепление вершины

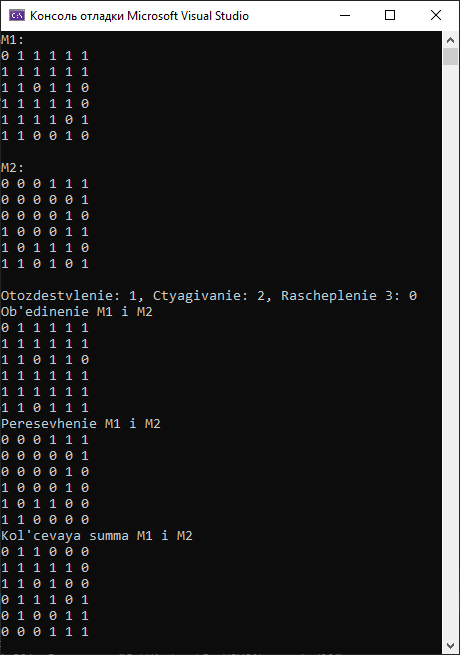


Рисунок 4 – Задание 3

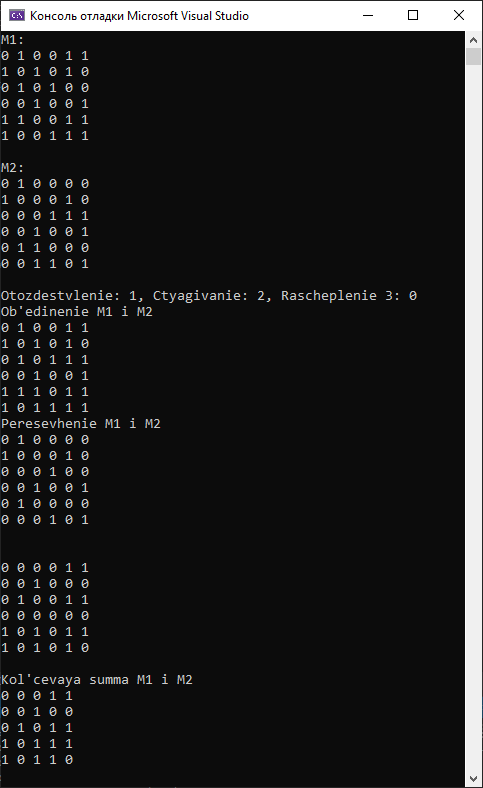


Рисунок 5 – Задание 3

Вывод

Изучил унарные и бинарные операции над графами, изучил методы работы с ними.