Федеральное государственное бюджетное образовательное учреждение высшего образования «Пензенский Государственный университет»

Кафедра «Вычислительная техника»

ОТЧЕТ

По лабораторной работе №6

по курсу «Логика и основы алгоритмизации в ИЗ»

на тему «Унарные и бинарные операции над графами»

Выполнили: ст. гр. 21ВВ1

Сурков А.А

Хорошильцев Е.А

Кривенков И.В

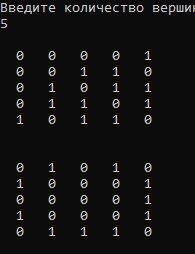
Приняли: Юрова О.В.

Акифьев И.В.

2022

**Задание 1**

1. Сгенерируйте (используя генератор случайных чисел) две матрицы *M*1*, М*2 смежности неориентированных помеченных графов *G*1, *G*2. Выведите сгенерированные матрицы на экран.



**Задание 2**

1. Для матричной формы представления графов выполните операцию:

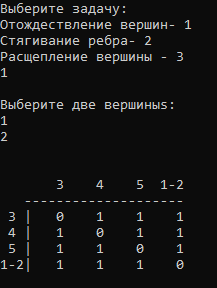
а) отождествления вершин

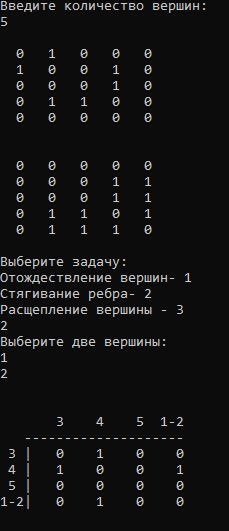
б) стягивания ребра

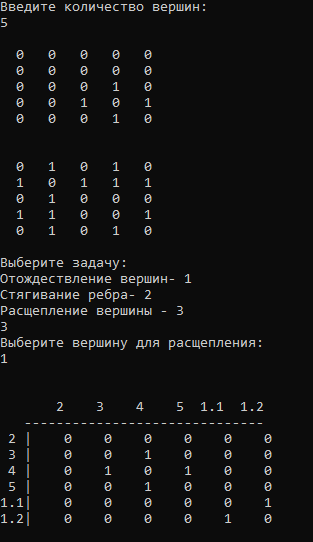
в) расщепления вершины

Номера выбираемых для выполнения операции вершин ввести с клавиатуры.

Результат выполнения операции выведите на экран.







**Задание 3**

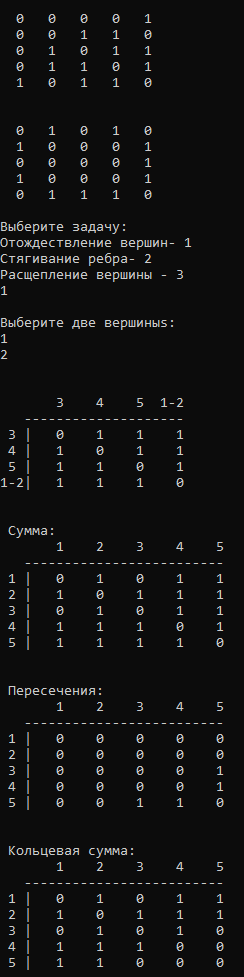
1. Для матричной формы представления графов выполните операцию:

а) объединения *G* = *G*1 https://lh6.googleusercontent.com/Oqk1O33n8UeZtm__lDco2Mzq-YFRLWgEkWee6dIftY6iZGkSGbz5uV9eNRTaVJkU_aHrajQpWS1ZhrSKl_bgEOXciEAg3Li-05E24kMqZBy4LeGYwDN3dvZu-tMzXnI0pFFsPVgmWvxEAQimFla9xxtXD3__s6vDDfYWKAV-wXxLy6mPu_fUF3s_K8FanaiqYM-DmoU9wQ *G*2

б) пересечения  *G* = *G*1 https://lh4.googleusercontent.com/0OF4D-R_1rtOIl9W-BVAViGb6QAJaRY5UGcKm-adiBg3ZlYIZlDzVZiyuKhpKppnoypdy7eOjB4zu3VHakawyEegC0QWX6xwq4eIYtaQN_8lz5IJ9TDmenempf2_ooOVfv18FY9OtjllUNnOfHOPs2SVzaTfrWT7zPJ1-gP0CtaUhzDfVHAmsJxCJsWJgomsPN1dgj5yRQ *G*2

в) кольцевой суммы *G* = *G*1 https://lh5.googleusercontent.com/NOc4UkAwdSuRIQsofSoyt5uYlwOtf9wu7BUaiDriFPJBzGfHrHKV6JLAiT4gc734OVqe3bZ0OxBJAgCxhujxrseYvwDW9ckv2dmxgGo1IMXrATBXAP6VJzFnH9tPuZf4jMYsY3ZtJr--DJh8-T8X_8vXEgh1ERQoSpF99fAD6Lx5zulDg_MrCU8EnjOkzPCxtA9KTMJfGQ *G*2

Результат выполнения операции выведите на экран.



**Вывод:**

Выполняя данную лабораторную, работу мы научились использовать унарные и бинарные операции над графами.

**Листинг программы**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "time.h"

#include "stdio.h"

#include "stdlib.h"

#include "malloc.h"

#include "locale.h"

int main(void)

{

setlocale(LC\_ALL, "Rus");

int m;

printf("Введите количество вершин: \n");

scanf("%d", &m);

int\*\* a, \*\* b;

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

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

srand(time(NULL));

// (1)

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

a[i] = (int\*)malloc(m \* sizeof(int));

b[i] = (int\*)malloc(m \* sizeof(int));

printf("\n");

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

a[i][j] = 0;

b[i][j] = 0;

if (j <= i) {

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

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

printf("%3d ", a[i][j]);

}

else {

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

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

printf("%3d ", a[i][j]);

}

}

}

//M2

printf("\n\n ");

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

b[i] = (int\*)malloc(m \* sizeof(int));

printf("\n");

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

b[i][j] = 0;

if (j <= i) {

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

printf("%3d ", b[i][j]);

}

else {

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

printf("%3d ", b[i][j]);

}

}

}

//-------------

// (2)

int operation;

printf("\n");

printf("\nВыберите задачу:\nОтождествление вершин- 1\nСтягивание ребра- 2\nРасщепление вершины - 3\n");

scanf("%d", &operation);

if (operation == 1) {

int peak1, peak2, repeatCheck = 0, k = m - 1, countI = 0, countJ = 0;

int\*\* task;

task = (int\*\*)malloc((k) \* sizeof(int\*));

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

task[i] = (int\*)malloc((k) \* sizeof(int));

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

task[i][j] = 0;

}

}

printf("\nВыберите две вершиныs:\n");

scanf("%d%d", &peak1, &peak2);

//---------- printing matrix

printf("\n\n ");

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

if (j != peak1 - 1 && j != peak2 - 1)

printf("%5d", (j + 1));

}

printf("%3d-%d", peak1, peak2);

printf("\n ");

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

printf("-----");

}

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

if (i != peak1 - 1 && i != peak2 - 1) {

countJ = 0;

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

if (j != peak1 - 1 && j != peak2 - 1) {

task[i - countI][j - countJ] = a[i][j];

}

else

countJ++;

}

task[i - countI][m - 2] = (a[peak1 - 1][i] + a[peak2 - 1][i]);

}

else countI++;

}

countI = 0;

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

if (i != peak1 - 1 && i != peak2 - 1) {

printf("\n %d |", (i + 1));

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

printf("%4d ", task[i - countI][j]);

}

}

else countI++;

}

countI = 0;

printf("\n%d-%d|", peak1, peak2);

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

if (i != peak2 - 1 && i != peak1 - 1)

task[m - 2][i - countI] = (a[peak1 - 1][i] + a[peak2 - 1][i]);

else

countI++;

}

task[m - 2][m - 2] = (a[peak1 - 1][peak1 - 1] + a[peak1 - 1][peak2 - 1]);

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

printf("%4d ", task[m - 2][j]);

}

}

else if (operation == 2) {

int peak1 = 1, peak2 = 1, repeatCheck = 0, k = m - 1, countI = 0, countJ = 0;

while (a[peak1 - 1][peak2 - 1] == 0) {

printf("Выберите две вершины:\n");

scanf("%d%d", &peak1, &peak2);

}

int\*\* task;

task = (int\*\*)malloc((k) \* sizeof(int\*));

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

task[i] = (int\*)malloc((k) \* sizeof(int));

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

task[i][j] = 0;

}

}

printf("\n\n ");

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

if (j != peak1 - 1 && j != peak2 - 1)

printf("%5d", (j + 1));

}

printf("%3d-%d", peak1, peak2);

printf("\n ");

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

printf("-----");

}

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

if (i != peak1 - 1 && i != peak2 - 1) {

countJ = 0;

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

if (j != peak1 - 1 && j != peak2 - 1) {

task[i - countI][j - countJ] = a[i][j];

}

else

countJ++;

}

task[i - countI][m - 2] = (a[peak1 - 1][i] + a[peak2 - 1][i]);

}

else countI++;

}

countI = 0;

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

if (i != peak1 - 1 && i != peak2 - 1) {

printf("\n %d |", (i + 1));

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

printf("%4d ", task[i - countI][j]);

}

}

else countI++;

}

countI = 0;

printf("\n%d-%d|", peak1, peak2);

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

if (i != peak2 - 1 && i != peak1 - 1)

task[m - 2][i - countI] = (a[peak1 - 1][i] + a[peak2 - 1][i]);

else

countI++;

}

task[m - 2][m - 2] = 0;

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

printf("%4d ", task[m - 2][j]);

}

}

else if (operation == 3)

{

int peak, k = m + 1, count = 2, countJ = 0, countI = 0;

printf("Выберите вершину для расщепления:\n");

scanf("%d", &peak);

int\*\* task;

task = (int\*\*)malloc((m + 1) \* sizeof(int\*));

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

task[i] = (int\*)malloc((m + 1) \* sizeof(int));

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

task[i][j] = 0;

}

}

printf("\n\n ");

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

if (j != peak - 1)

printf("%5d", (j + 1));

}

printf("%3d.1%3d.2", peak, peak);

printf("\n ");

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

printf("-----");

}

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

if (i != peak - 1) {

countJ = 0;

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

if (j != peak - 1) {

task[i - countI][j - countJ] = a[i][j];

}

else

countJ++;

}

if (a[i][peak - 1] == 0) {

task[i - countI][m - 1] = 0;

task[i - countI][m] = 0;

}

else if (count % 2 == 0) {

task[i - countI][m - 1] = 1;

task[i - countI][m] = 0;

count++;

}

else if (count % 2 == 1) {

task[i - countI][m - 1] = 0;

task[i - countI][m] = 1;

count++;

}

}

else countI = 1;

}

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

if (i >= peak - 1) {

printf("\n %d |", (i + 2));

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

printf("%5d", task[i][j]);

}

}

else {

printf("\n %d |", (i + 1));

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

printf("%5d", task[i][j]);

}

}

}

printf("\n%d.1|", peak);

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

task[m - 1][j] = task[j][m - 1];

task[m][j] = task[j][m];

}

task[m - 1][m - 1] = 0;

task[m - 1][m] = 1;

task[m][m - 1] = 1;

task[m][m] = 0;

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

printf("%5d", task[m - 1][j]);

}

printf("\n%d.2|", peak);

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

printf("%5d", task[m][j]);

}

}

//------------------

// (3)

int\*\* Sum, \*\* Mul, \*\* RSum;

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

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

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

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

Sum[i] = (int\*)malloc(m \* sizeof(int));

Mul[i] = (int\*)malloc(m \* sizeof(int));

RSum[i] = (int\*)malloc(m \* sizeof(int));

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

Sum[i][j] = 0;

Mul[i][j] = 0;

RSum[i][j] = 0;

}

}

printf("\n\n\n Сумма:\n");

printf(" ");

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

printf("%5d", (j + 1));

}

printf("\n ");

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

printf("-----");

}

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

printf("\n %d |", (i + 1));

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

Sum[i][j] = (a[i][j] + b[i][j]);

if (Sum[i][j] == 2) Sum[i][j] = 1;

printf("%4d ", Sum[i][j]);

}

}

printf("\n\n\n Пересечения:\n");

printf(" ");

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

printf("%5d", (j + 1));

}

printf("\n ");

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

printf("-----");

}

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

printf("\n %d |", (i + 1));

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

Mul[i][j] = (a[i][j] \* b[i][j]);

printf("%4d ", Mul[i][j]);

}

}

printf("\n\n\n Кольцевая сумма:\n");

printf(" ");

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

printf("%5d", (j + 1));

}

printf("\n ");

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

printf("-----");

}

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

printf("\n %d |", (i + 1));

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

if (a[i][j] + b[i][j] == 1)

RSum[i][j] = 1;

else

RSum[i][j] = 0;

printf("%4d ", RSum[i][j]);

}

}

free(a);

free(b);

getchar();

getchar();

return(0);

}