Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Системы автоматизированного проектирования»

**Отчет**

По лабораторной работе №3

По курсу «Логика и основы алгоритмизации в инженерных задачах»

На тему «Унарные и бинарные операции над графами»

Выполнил студент гр.20ВВ3

Мирясов Н. А.

Проверили:

Митрохин М. А.

Юрова О. В.

Пенза, 2021

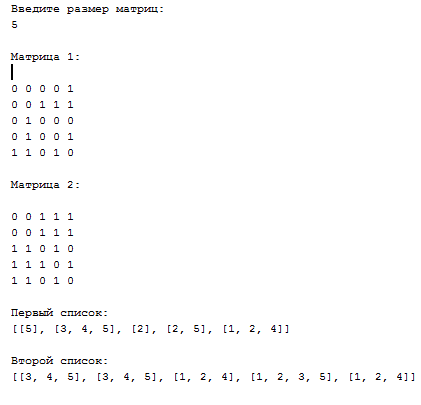
**Цель:** научится производить унарные и бинарные операции с матрицами смежности.

**Задание 1:**

1. Сгенерировали (используя генератор случайных чисел) две матрицы M1, М2 смежности неориентированных помеченных графов G1, G2.

2. \* Для указанных графов преобразовали представление матриц смежности в списки смежности.

Вывели сгенерированные матрицы и полученные списки на экран:



**Задание 2**

1. Для матричной формы представления графов выполнили операцию:

а) отождествления вершин

б) стягивания ребра

в) расщепления вершины

Номера выбираемых для выполнения операции вершин ввели с клавиатуры.

1. \* Для представления графов в виде списков смежности выполнили операцию:

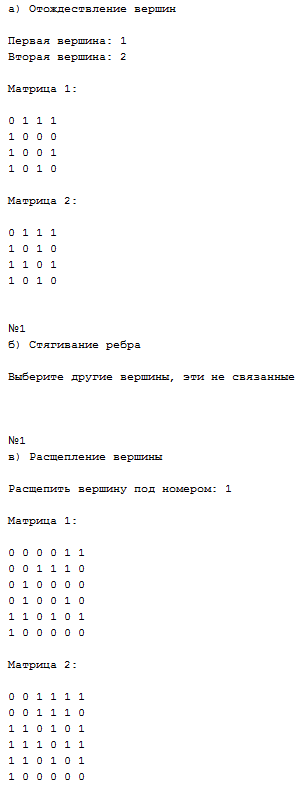
а) отождествления вершин

б) стягивания ребра

в) расщепления вершины

Номера выбираемых для выполнения операции вершин ввели с клавиатуры.

Результат выполнения операций:



**Задание 3**

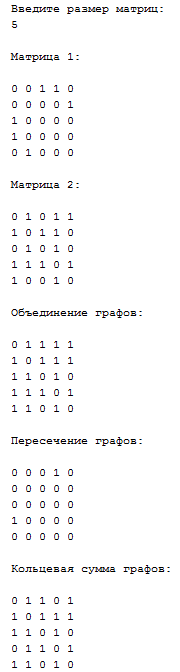
1. Для матричной формы представления графов выполнили операцию:

а) объединения *G* = *G*1*G*2

б) пересечения *G* = *G*1*G*2

в) кольцевой суммы *G* = *G*1*G*2

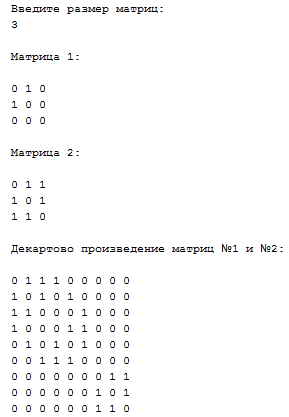
Результат выполнения операции:



**Задание 4 \***

1. Для матричной формы представления графов выполнили операцию декартова произведения графов *G = G*1X*G*2.

Результат выполнения операции:



**Листинг задания 1-2:**

package lb3;

import java.util.\*;

public class mission1 {

public static void main(String[] args) {

System.out.println("Введите размер матриц:");

Scanner in = new Scanner(System.in);

int n = in.nextInt();

int[][] matrix1 = new int[n][n];

int[][] matrix2 = new int[n][n];

int[][] matrix3 = new int[n + 1][n + 1];

int[][] matrix4 = new int[n + 1][n + 1];

int[][] copymatrix1 = new int[n][n];

int[][] copymatrix2 = new int[n][n];

ArrayList<LinkedList<Integer>> adjLists1 = new ArrayList<LinkedList<Integer>>();

ArrayList<LinkedList<Integer>> adjLists2 = new ArrayList<LinkedList<Integer>>();

rand(n, matrix1, matrix2);

addEdge(n, matrix1, matrix2, adjLists1, adjLists2);

output(n, matrix1, matrix2, adjLists1, adjLists2);

//---------------------------------------Задание 2

System.out.println("\n\n№1\na) Отождествление вершин");

System.out.print("\nПервая вершина: ");

Scanner in1 = new Scanner(System.in);

int vertex1 = in1.nextInt() - 1;

System.out.print("Вторая вершина: ");

Scanner in2 = new Scanner(System.in);

int vertex2 = in2.nextInt() - 1;

// №1

for (int i = 0; i < matrix1.length; i++)

for (int j = 0; j < matrix1[i].length; j++) {

copymatrix1[i][j] = matrix1[i][j];

copymatrix2[i][j] = matrix2[i][j];

}

identification(n, vertex1, vertex2, copymatrix1, copymatrix2);

constriction(n, vertex1, vertex2, copymatrix1, copymatrix2, matrix1, matrix2);

for (int i = 0; i < matrix1.length; i++)

for (int j = 0; j < matrix1[i].length; j++) {

matrix3[i][j] = matrix1[i][j];

matrix4[i][j] = matrix2[i][j];

}

separation(matrix3, matrix4);

// №2

}

public static void rand(int n, int[][] matrix1, int[][] matrix2) {

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

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

matrix1[i][j] = (int) ((Math.random() \* 2) + 0);

matrix1[j][i] = matrix1[i][j];

matrix2[i][j] = (int) ((Math.random() \* 2) + 0);

matrix2[j][i] = matrix2[i][j];

if (i == j) {

matrix1[i][j] = 0;

matrix2[i][j] = 0;

}

}

}

}

public static void addEdge(int n, int[][] matrix1, int[][] matrix2, ArrayList<LinkedList<Integer>> adjLists1, ArrayList<LinkedList<Integer>> adjLists2) {

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

adjLists1.add(new LinkedList<Integer>());

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

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

if (matrix1[i][j] == 1)

adjLists1.get(i).add(j + 1);

}

}

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

adjLists2.add(new LinkedList<Integer>());

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

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

if (matrix2[i][j] == 1)

adjLists2.get(i).add(j + 1);

}

}

}

public static void output(int n, int[][] matrix1, int[][] matrix2, ArrayList<LinkedList<Integer>> adjLists1, ArrayList<LinkedList<Integer>> adjLists2) {

System.out.println("\nМатрица 1:\n");

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

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

System.out.print(matrix1[i][j] + " ");

}

System.out.println();

}

System.out.println("\nМатрица 2:\n");

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

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

System.out.print(matrix2[i][j] + " ");

}

System.out.println();

}

System.out.println("\nПервый список:");

System.out.println(adjLists1);

System.out.println("\nВторой список:");

System.out.println(adjLists2);

}

//---------------------------------------Задание 2

// №1 - Отождествление вершин

public static void identification(int n, int vertex1, int vertex2, int[][] matrix1, int[][] matrix2) {

int max;

int min;

int[] matrix3 = new int[n];

int[] matrix4 = new int[n];

if (vertex1 > vertex2) {

max = vertex1;

min = vertex2;

} else {

max = vertex2;

min = vertex1;

}

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

matrix3[i] = matrix1[min][i] + matrix1[max][i];

matrix4[i] = matrix2[min][i] + matrix2[max][i];

if (matrix3[i] == 2)

matrix3[i] = 1;

if (matrix4[i] == 2)

matrix4[i] = 1;

}

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

matrix1[min][i] = matrix3[i];

matrix1[i][min] = matrix3[i];

matrix2[min][i] = matrix4[i];

matrix2[i][min] = matrix4[i];

}

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

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

if (i >= max) {

matrix1[i][j] = matrix1[i + 1][j];

matrix2[i][j] = matrix2[i + 1][j];

} else if (j >= max) {

matrix1[i][j] = matrix1[i][j + 1];

matrix2[i][j] = matrix2[i][j + 1];

}

if (i >= max & j >= max) {

matrix1[i][j] = matrix1[i + 1][j + 1];

matrix2[i][j] = matrix2[i + 1][j + 1];

}

}

}

System.out.println("\nМатрица 1:\n");

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

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

System.out.print(matrix1[i][j] + " ");

}

System.out.println();

}

System.out.println("\nМатрица 2:\n");

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

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

System.out.print(matrix2[i][j] + " ");

}

System.out.println();

}

}

// №1 - Стягивание ребра

public static void constriction(int n, int vertex1, int vertex2, int[][] copymatrix1, int[][] copymatrix2, int[][] matrix1, int[][] matrix2) {

System.out.println("\n\n№1\nб) Стягивание ребра");

if (matrix1[vertex1][vertex2] != 1 & matrix2[vertex1][vertex2] != 1) {

System.out.println("\nВыберите другие вершины, эти не связанные\n");

return;

}

if (matrix1[vertex1][vertex2] == 1) {

System.out.println("\nМатрица 1:\n");

for (int i = 0; i < copymatrix1.length - 1; i++) {

for (int j = 0; j < copymatrix1.length - 1; j++) {

if (i == j) {

copymatrix1[i][j] = 0;

}

System.out.print(copymatrix1[i][j] + " ");

}

System.out.println();

}

}

if (matrix2[vertex1][vertex2] == 1) {

System.out.println("\nМатрица 2:\n");

for (int i = 0; i < copymatrix2.length - 1; i++) {

for (int j = 0; j < copymatrix2.length - 1; j++) {

if (i == j) {

copymatrix2[i][j] = 0;

}

System.out.print(copymatrix2[i][j] + " ");

}

System.out.println();

}

}

}

// №1 - Расщепление вершины

public static void separation(int[][] matrix1, int[][] matrix2){

System.out.println("\n\n№1\nв) Расщепление вершины");

System.out.print("\nРасщепить вершину под номером: ");

Scanner in1 = new Scanner(System.in);

int vertex3 = in1.nextInt() - 1;

for (int i = 0; i < matrix1.length - 1; i++) {

matrix1[i][matrix1.length - 1] = matrix1[i][vertex3];

matrix2[i][matrix2.length - 1] = matrix2[i][vertex3];

}

matrix1[vertex3][matrix1.length - 1] = matrix1[matrix1.length - 1][vertex3] = 1;

matrix1[matrix1.length - 1][matrix1.length - 1] = 0;

matrix2[vertex3][matrix2.length - 1] = matrix2[matrix2.length - 1][vertex3] = 1;

matrix2[matrix2.length - 1][matrix2.length - 1] = 0;

System.out.println("\nМатрица 1:\n");

for (int i = 0; i < matrix1.length; i++) {

for (int j = 0; j < matrix1[i].length; j++) {

System.out.print(matrix1[i][j] + " ");

}

System.out.println();

}

System.out.println("\nМатрица 2:\n");

for (int i = 0; i < matrix2.length; i++) {

for (int j = 0; j < matrix2[i].length; j++) {

System.out.print(matrix2[i][j] + " ");

}

System.out.println();

}

}

}

**Листинг задания 3:**

package lb3;

import java.util.\*;

public class mission3 {

public static void main(String[] args) {

System.out.println("Введите размер матриц:");

Scanner in = new Scanner(System.in);

int n = in.nextInt();

int[][] matrix1 = new int[n][n];

int[][] matrix2 = new int[n][n];

mission1 rnd = new mission1();

rnd.rand(n, matrix1, matrix2);

output(n, matrix1, matrix2);

//----------------------- Задание 3

int[][] matrix3 = new int[n][n];

combineGraphs(n, matrix1, matrix2, matrix3);

confluence(n, matrix1, matrix2, matrix3);

annularSum(n, matrix1, matrix2, matrix3);

}

public static void output(int n, int[][] matrix1, int[][] matrix2) {

System.out.println("\nМатрица 1:\n");

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

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

System.out.print(matrix1[i][j] + " ");

}

System.out.println();

}

System.out.println("\nМатрица 2:\n");

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

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

System.out.print(matrix2[i][j] + " ");

}

System.out.println();

}

}

//---------------------------------------Задание 3

// №1 - Объединение графов

public static void combineGraphs(int n, int[][] matrix1, int[][] matrix2, int[][] matrix3){

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

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

if (matrix1[i][j] == 1 | matrix2[i][j] == 1)

matrix3[i][j] = 1;

else

matrix3[i][j] = 0;

}

}

System.out.println("\nОбъединение графов:\n");

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

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

System.out.print(matrix3[i][j] + " ");

}

System.out.println();

}

}

// №2 - Пересечение графов

public static void confluence(int n, int[][] matrix1, int[][] matrix2, int[][] matrix3){

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

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

if (matrix1[i][j] == 1 & matrix2[i][j] == 1)

matrix3[i][j] = 1;

else

matrix3[i][j] = 0;

}

}

System.out.println("\nПересечение графов:\n");

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

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

System.out.print(matrix3[i][j] + " ");

}

System.out.println();

}

}

// №3 - Кольцевая сумма графов

public static void annularSum(int n, int[][] matrix1, int[][] matrix2, int[][] matrix3){

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

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

if ((matrix1[i][j] == 1 & matrix2[i][j] == 0) | (matrix1[i][j] == 0 & matrix2[i][j] == 1))

matrix3[i][j] = 1;

else

matrix3[i][j] = 0;

}

}

System.out.println("\nКольцевая сумма графов:\n");

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

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

System.out.print(matrix3[i][j] + " ");

}

System.out.println();

}

}

}

**Листинг задания 4:**

package lb3;

import java.util.\*;

public class mission4 {

public static void main(String[] args) {

System.out.println("Введите размер матриц:");

Scanner in = new Scanner(System.in);

int n = in.nextInt();

int[][] matrix1 = new int[n][n];

int[][] matrix2 = new int[n][n];

mission1 rnd = new mission1();

rnd.rand(n, matrix1, matrix2);

output(n, matrix1, matrix2);

//----------------------- Задание 4

int[][] matrix3 = new int[n\*n][n\*n];

Decart(n, matrix1, matrix2, matrix3);

}

public static void output(int n, int[][] matrix1, int[][] matrix2) {

System.out.println("\nМатрица 1:\n");

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

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

System.out.print(matrix1[i][j] + " ");

}

System.out.println();

}

System.out.println("\nМатрица 2:\n");

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

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

System.out.print(matrix2[i][j] + " ");

}

System.out.println();

}

}

//---------------------------------------Задание 4

// №1 - Декартово произведение графов

public static void Decart(int n, int[][] matrix1, int[][] matrix2, int[][] matrix3){

int count = -1;

for (int i = 0; i < matrix1.length; i++) {

for (int j = 0; j < matrix2.length; j++) {

count += 1;

int number = 0;

for ( int ii = 0; ii < matrix1.length; ii++){

for (int jj = 0; jj < matrix2.length; jj++){

if(i == jj & j == ii)

matrix3[count][number] = 0;

else if(i == ii)

matrix3[count][number] = matrix2[j][jj];

else if (j == jj)

matrix3[count][number] = matrix1[i][ii];

else

matrix3[count][number] = 0;

number += 1;

}

}

}

}

System.out.println("\nДекартово произведение матриц №1 и №2:\n");

for (int i = 0; i < n\*n; i++) {

for (int j = 0; j < n\*n; j++) {

System.out.print(matrix3[i][j] + " ");

}

System.out.println();

}

}

}

**Вывод:** научились производить унарные и бинарные операции с матрицами смежности.