Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В. Ф. Уткина»

Кафедра «САПР»

Дисциплина

«Дискретная математика»

Отчёт по лабораторной работе №2

по теме:

«ПОИСК КРАТЧАЙШИХ ПУТЕЙ НА ГРАФАХ»

**Выполнили:**

ст. гр. 135

Бардин М.С.

Кузнецова М.А.

**Проверил**

Доц. Орешков В. И.

Доц. Бакулева М. А.

Рязань 2023

**Цель работы**

Изучение алгоритмов поиска кратчайших путей на графах на примере метода динамического программирования.

**Задание**

Создать программу, реализующую метод динамического программирования и алгоритм топологической сортировки вершин. Исходный граф задаётся в виде матрицы смежности.

Вариант 11 (рисунок 1):

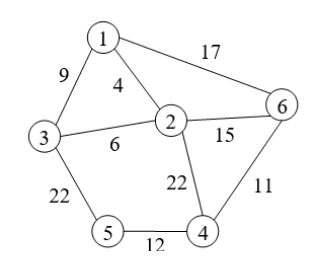


Рисунок 1 – Исходный граф

Блок схема алгоритма представлена на рисунках 2, 3, 4



Рисунок 2 - БСА 1 часть

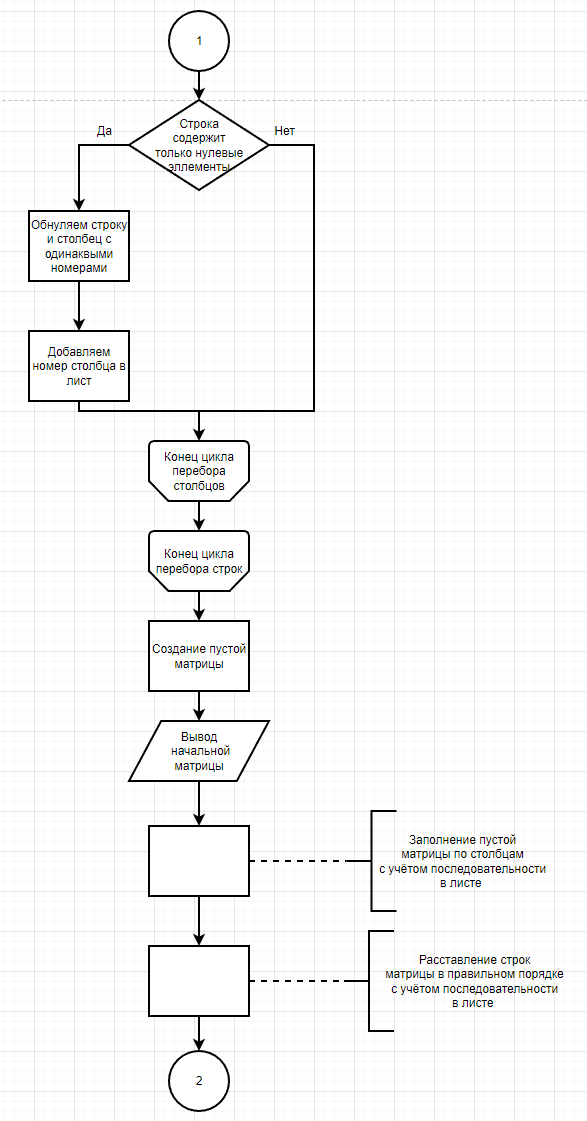


Рисунок 3 - БСА часть 2

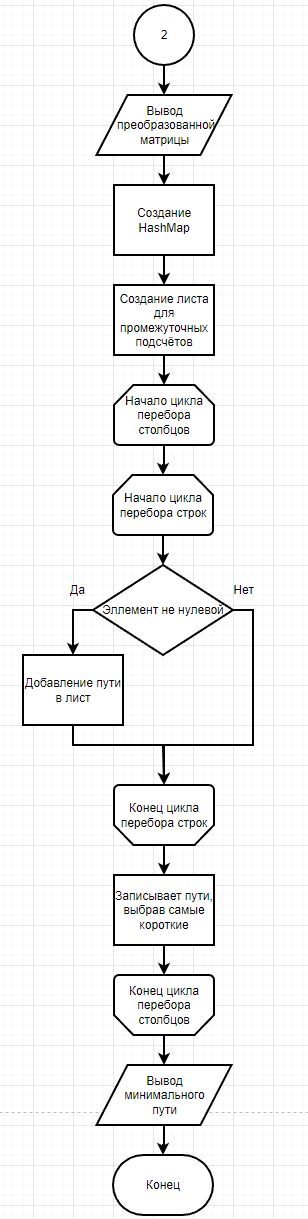


Рисунок 4 - БСА часть 3

**Текст программы (Java):**

import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.HashMap;  
import java.util.List;  
  
public class Main {  
  
 public static void printMatrix(int[][] matrix){  
 for (int i = 0; i < matrix.length; i++) {  
 for (int j = 0; j < matrix.length; j++) {  
 System.out.print(matrix[i][j] + " ");  
  
 }  
 System.out.println();  
 }  
 }  
  
 public static int[][] copyMatrix(int[][] matrix){  
 int[][] returnMatrix = new int[matrix.length][matrix.length];  
 for (int i = 0; i < matrix.length; i++) {  
 for (int j = 0; j < matrix.length; j++) {  
 returnMatrix[i][j]=matrix[i][j];  
 }  
 }  
 return returnMatrix;  
 }  
 public static void main(String[] args) {  
  
 int[][] matrix1 = {   
{0, 4, 9, 0, 0, 17},  
{0, 0, 0, 22, 0, 15},  
{0, 6, 0, 0, 22, 0},  
{0, 0, 0, 0, 0, 11},  
{0, 0, 0, 12, 0, 0},  
{0, 0, 0, 0, 0, 0}

};  
  
  
 int[][] matrix2 = copyMatrix(matrix1);  
  
 List<Integer> list = new ArrayList<>();  
  
 for (int i = 0; i < matrix1.length; i++) {  
 for (int j = 0; j < matrix1.length; j++) {  
 if (list.contains(j)) {  
 continue;  
 }  
 if (Arrays.stream(matrix1[j]).allMatch(s -> s == 0)) {  
 for (int k = 0; k < matrix1.length; k++) {  
 matrix1[j][k] = 0;  
 matrix1[k][j] = 0;  
 }  
 printMatrix(matrix1);  
  
 list.add(j);  
 System.out.println(list);  
 }  
 }  
  
 }  
  
 int[][] matrix = new int[matrix1.length][matrix1.length];  
  
 System.out.println("Начальная матрица:");  
 printMatrix(matrix2);  
 System.out.println();  
  
 for (int i = 0; i < matrix.length; i++) {  
 for (int j = 0; j < matrix.length; j++) {  
 matrix[j][i] = matrix2[j][list.get(matrix.length-(i+1))];  
  
 }  
 printMatrix(matrix);  
 System.out.println();  
  
 }  
  
 int[][] matrix3 = copyMatrix(matrix);  
  
 for (int i = 0; i < matrix1.length; i++) {  
 matrix[i]= matrix3[list.get(matrix.length-(i+1))] ;  
  
 }  
  
 System.out.println();  
  
 System.out.println("Преобразованная матрица:");  
 printMatrix(matrix);  
  
 HashMap<Integer,Integer> hashMap = new HashMap<>();  
 hashMap.put(0,0);  
  
  
 for (int i = 1; i < matrix.length; i++) {  
 List<Integer> value = new ArrayList<>();  
 for (int j = 0; j < matrix.length; j++) {  
 if (matrix[j][i]!=0){  
 value.add(matrix[j][i]+hashMap.get(j));  
 }  
   
 }  
  
 hashMap.put(i,value.stream().min(Integer::compare).get());  
 }  
  
 System.out.println();  
 System.out.println("Кратчайший путь");  
 System.out.println(hashMap.get(matrix.length-1));  
 }  
}

**Контрольный пример (рисунок 5):**

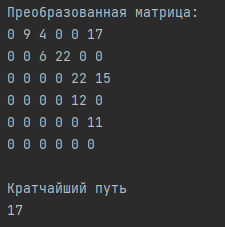


Рисунок 5 - Топологическая сортировка и нахождение минимального пути с помощью метода динамического программирования

**Вывод**

Мы изучили алгоритмы поиска кратчайших путей на графах на примере метода динамического программирования. Научились реализовывать метод топологической сортировки на языке Java.