Министерство образования и науки Российской Федерации

ФГБОУ ВО «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

Лабораторная работа №4

«Графы»

Выполнил студент

группы ИВТАПбд-11

Мухаметзянов Т. А.

Ульяновск, 2022

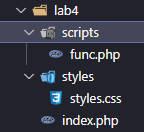
**Постановка задачи**

Разработать программу на php нахождения кратчайшего пути в графе между всеми парами вершин. Вид графа определяется заданием. Алгоритм определяется самостоятельно

**Реализация**

Структура

1. PHP страница для пользователя
2. PHP файл с основным скриптом для корректной работы страницы
3. CSS файл со всеми стилями



Лабораторная работа реализована в 3 файла (php страница, css, php). На сайте пользователь вводит количество вершин графа, а также сами элементы графа через пробел.

1. Если ввод не проходит валидацию, тогда пользователю выводит ошибку, которую необходим исправить
2. Если ввод корректен, тогда пользователь получает введенную матрицу, матрицу конечных путей и матрицу всех маршрутов от 1-ой до последней вершины.

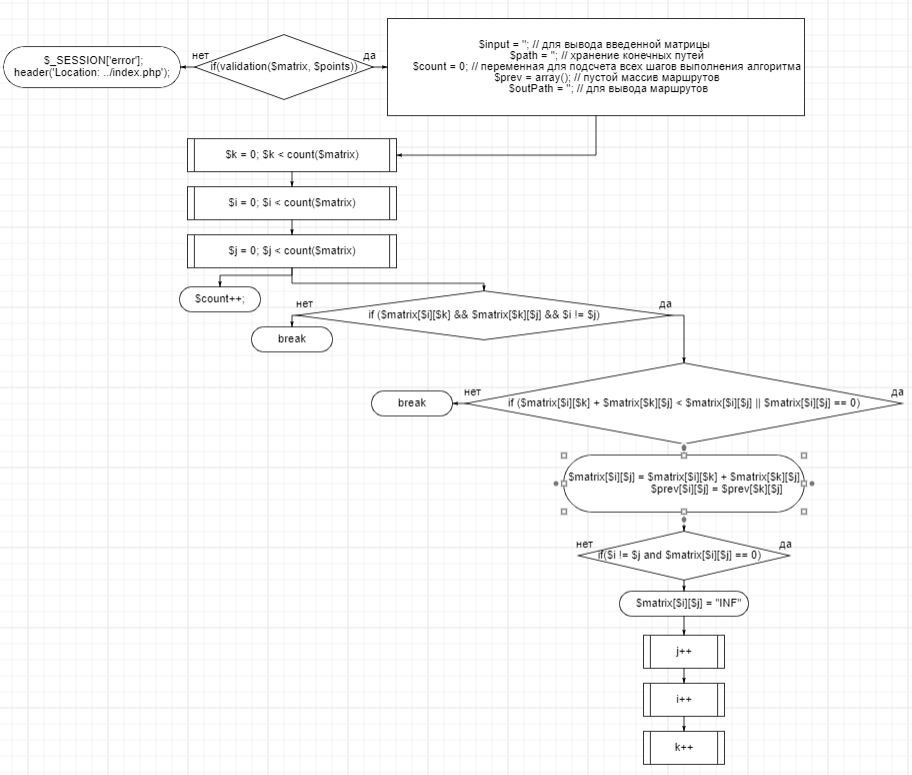
Для отображения матрицы конечных путей и матрицы маршрутов использован алгоритм Флойда-Уоршелла.

Если путь отсутствует значение 0 меняется на INF. Также алгоритм записывает количество всех шагов его выполнения.

Сайт состоит из номера л\р, а также варианта, полей ввода количества вершин графа и самого графа, кнопки подтверждения (связанная с основной функцией на PHP) и абзацев для вывода результатов.

**Блок – схема**

Алгоритм Флойда-Уоршелла



**Интерфейс работы**

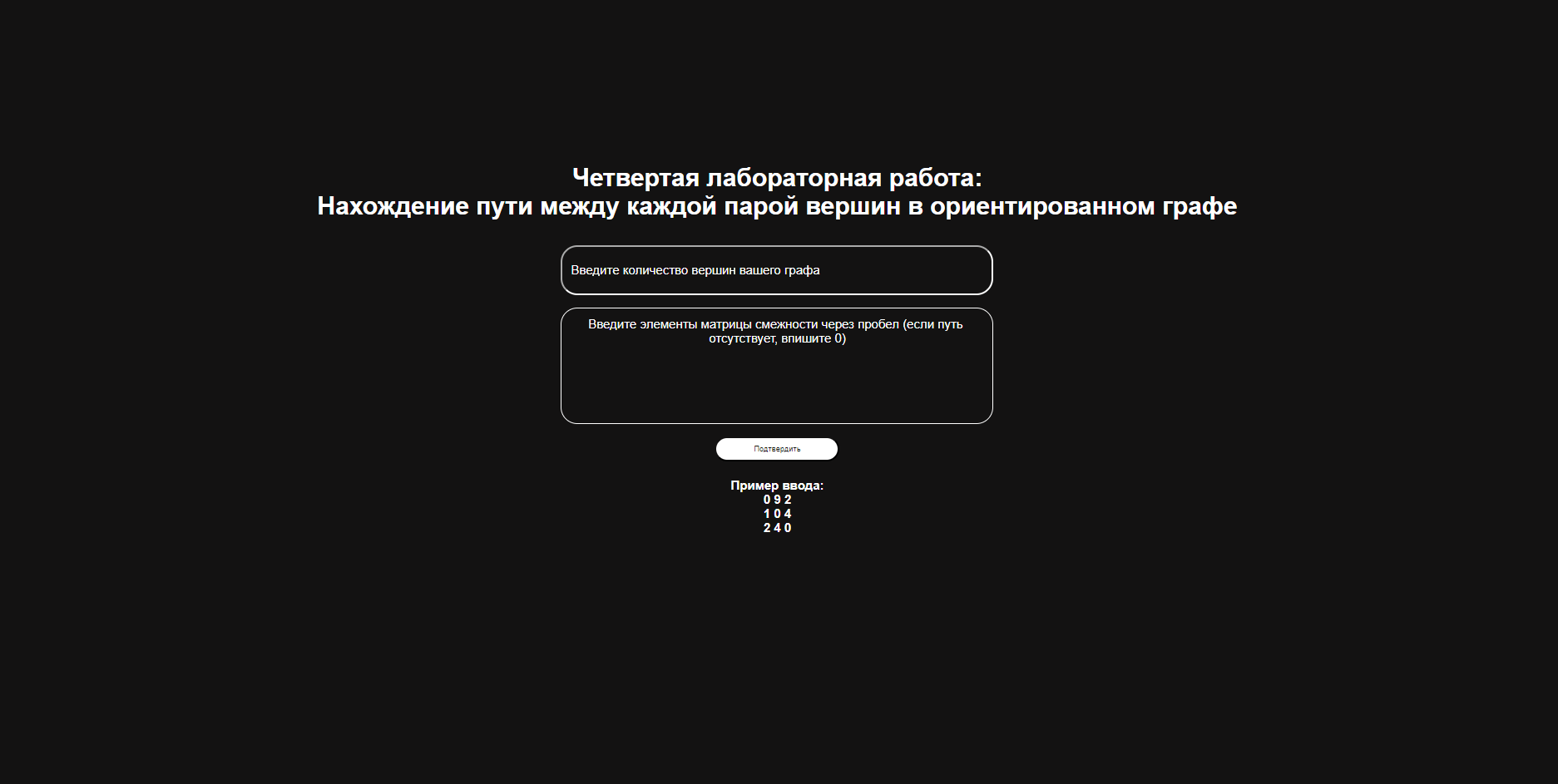


Рис 1 – Вид страницы



Рис 2 – Ошибка валидации

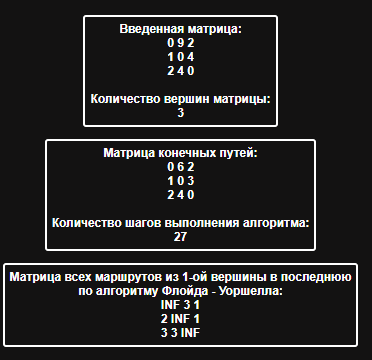


Рис 3 – Результаты

**Листинг кода**

PHP

<?php

    session\_start();

    // считывание матрицы смежности && преобразование строки в массив

    $matrix = preg\_split('/[\n\r]+/', $\_POST['array']);

    for ($i = 0; $i < count($matrix); $i++) {

        $matrix[$i] = trim($matrix[$i]);

        $matrix[$i] = preg\_replace('/\s+/', ' ', $matrix[$i]);

        $matrix[$i] = explode(" ", $matrix[$i]);

    }

    // считывание кол-ва вершин графа

    $points = trim(($\_POST['firstTop']), " ");

    //функция валидации

    function validation($matrix, $points) {

        $\_SESSION['text'] = "";

        if(strlen($points) == 0 or count($matrix) == 0) {

            $\_SESSION['text'] = "Не должно быть ни одного пустого поля!";

            return false;

        }

        for ($i = 0; $i < count($matrix); $i++){

            if (count($matrix) != count($matrix[$i])) {

                $\_SESSION['text'] = "Матрица должна быть квадратной!";

                return false;

            }

            for ($j = 0; $j < count($matrix); $j++){

                if($i == $j and $matrix[$i][$j] != 0) {

                    $\_SESSION['text'] = "На главной диагонали должны быть нули!";

                    return false;

                }

                if(!is\_numeric($matrix[$i][$j])) {

                    $\_SESSION['text'] = "Матрица должна состоять из цифр!";

                    return false;

                }

            }

        }

        if (!is\_numeric($points)) {

            $\_SESSION['text'] = "Кол-во вершин должно состоять из цифр!";

            return false;

        }

        if ($points != count($matrix)) {

            $\_SESSION['text'] = "Число вершин не сходится с графом!";

            return false;

        }

        return true;

    }

    if(validation($matrix, $points)) {

        $input = ''; // для вывода введенной матрицы

        $path = ''; // хранение конечных путей

        $count = 0; // переменная для подсчета всех шагов выполнения алгоритма

        $prev = array(); // пустой массив маршрутов

        $outPath = ''; // для вывода маршрутов

        //вывод введенной матрицы смежности

        for ($i = 0; $i < count($matrix); $i++) {

            for ($j = 0; $j < count($matrix); $j++) {

                $input = $input.$matrix[$i][$j]." ";

            }

            $input = $input."<br>";

        }

        // инициалиацзия массива маршрутов от 1 до последней вершины

        for ($i = 0; $i < count($matrix); $i++) {

            for ($j = 0; $j < count($matrix); $j++) {

                $prev[$i][$j] = $i + 1;

            }

            $prev[$i][$j] = -1;

        }

        //алгорит Флойда-Уоршелла для нахождения минимального пути между каждой парой элементов

        for ($k = 0; $k < count($matrix); $k++) {

            for ($i = 0; $i < count($matrix); $i++) {

                for ($j = 0; $j < count($matrix); $j++) {

                    $count++;

                    if ($matrix[$i][$k] && $matrix[$k][$j] && $i != $j) {

                        if ($matrix[$i][$k] + $matrix[$k][$j] < $matrix[$i][$j] || $matrix[$i][$j] == 0) {

                            $matrix[$i][$j] = $matrix[$i][$k] + $matrix[$k][$j]; // конечные пути

                            $prev[$i][$j] = $prev[$k][$j]; // заносим в массив маршрутов сами маршруты

                        }

                    }

                }

            }

        }

        // замена всех 0 не находящихся на главной диагонали на бесконечность (INF)

        for ($i = 0; $i < count($matrix); $i++) {

            for ($j = 0; $j < count($matrix); $j++) {

                if($i != $j and $matrix[$i][$j] == 0) {

                    $matrix[$i][$j] = "INF"; // отсутствие пути

                }

                $path = $path.$matrix[$i][$j]." ";

            }

            $path = $path."<br>";

        }

        // вывод всех маршрутов

        for ($i = 0; $i < count($prev); $i++) {

            for ($j = 0; $j < count($prev); $j++) {

                if($i == $j) {

                    $prev[$i][$j] = "INF"; // отсутствие маршрута

                }

                $outPath = $outPath.$prev[$i][$j]." ";

            }

            $outPath = $outPath."<br>";

        }

        $\_SESSION['matrix'] = "Введенная матрица:<br>" . $input. "<br>Количество вершин матрицы:<br>" . count($matrix). "<br>";

        $\_SESSION['final'] = "Матрица конечных путей:<br>" . $path. "<br>Количество шагов выполнения алгоритма:<br>" . $count . "<br>";

        $\_SESSION['let'] = "Матрица всех маршрутов из 1-ой вершины в последнюю<br> по алгоритму Флойда - Уоршелла:<br>" . $outPath. "";

        header('Location: ../index.php');

    }

    else {

        header('Location: ../index.php');

    }

?>

**Выводы**

С помощью данной лабораторной работы были изучены построение матрицы кратчайших путей между парами всех вершин и матрицы маршрутов всех пар вершин по алгоритму Флойда-Уоршелла.

**Литература**

1. Р. Хаггарти. Дискретная математика для программистов.
2. https://myslide.ru/presentation/skachatalgoritmizaciyaiprogrammirovanieyazykCcelochislennyealgoritmy3845
3. https://kvodo.ru/algoritm-floyda-uorshella.html