Министерство образования и науки Российской Федерации

Ульяновский Технический университет

Кафедра: вычислительная техника

Дисциплина: Дискретная математика

Лабораторная работа №5.

«Матрица достижимости»

Выполнил:

Студент: ИВТАП Бд-11

Кондратьев Павел Сергеевич

Проверил:

Игонин Андрей Геннадьевич

Ульяновск, 2017

Содержание

|  |  |
| --- | --- |
| 1. Цели и задачи……………………………………………………………… | 3 |
| 2. Ход выполнения работы…………………………………………………. | 3 |
| 3. Выводы ……………………………………………………………………. | 4 |
| 4. Список используемой литературы…...…………………………………... | 4 |
| 5. Приложение №1…………………………………………………………… | 5 |

**Цели и задачи**

**Цель**

Развитие, расширение и понимание устройства Интернета, с помощью изучения языков программирования, которые используются в создании сайта. Изучение языка программирования PHP и его использование в различных задачах.

**Задача**

Разработать программу на php которая находит Матрицу достижимости. Ввод матрицы (двоичная матрица) осуществляется пользователем в отведенном для этого месте (Форма ввода данных).

**Ход выполнения работы**

Для реализации данной задачи нам потребуется:

1. Создать форму для ввода данных
2. Разработать функции сложение и умножение матриц
3. Реализовать функцию нахождения Матрицы Достижимости

Для создания формы ввода, потребуется 2 тега <input> в самой <form>, которые отвечают за ввод данных (для этого указываем: type="text") и за действие над матрицей. Также потребуется тэг <textarea> для ввода двоичной матрицы.

<form method="post" action="dost.php">  
 Матрица достижимости<br>  
 <input name="size" type="text" placeholder="n"><br>  
 Введите матрицу nxn:<br><textarea style="resize: none" rows="20" cols="50" name="matrix">**<?=** htmlspecialchars($\_POST['matrix']) **?>**</textarea><br>  
 <input type="submit" style="width: 300px;margin: 25px" value="Do it">  
 <br>**<?php echo** "n = ", $\_POST['size']; **?>**</form>

После того как фора готова, прописываем валидацию для ввода матрицы, которая проверяет то что матрица является квадратной и имеет бинарный вид.

**Матрица достижимости -** таблица маршрутизации, в которой указываются только те узлы, с которыми данный узел поддерживает связь. Обычно используется для контроля образования циклов в маршрутах.

Нахождение матрицы достижимости E^\* сведётся к обычному пошаговому перемножению матриц с последующим сложением результатов каждого шага. Тогда получившаяся матрица E^\* будет состоять не только из 0 и 1 и будет характеризовать не только наличие путей между вершинами, но уже и количество таких путей. В таком случае можно разрешить наличие кратных рёбер в графе.

Для реализации этого алгоритма мы напишем 2 функции (сложение и умножение матриц). Умножение матриц производиться путем умножения строки на столбец этой матрицы. Далее создаем цикл, в котором сначала умножаем введенную матрицу после чего складываем с тем что было и так n раз (n – размерность матрицы). При первом проходе в цикле, мы умножаем введенную матрицу на единичную матрицу (у которой на главной диагонали 1).

**for** ($i = 0; $i < *SIZE*; $i++) {  
 $new\_matrix = matrix\_multiply($new\_matrix, $elem);  
 $res = matrix\_sum($res, $new\_matrix);  
}

Вся программа сводиться к этому циклу, в котором идет перемножение матрицы и последующее ее сложение.

**Вывод**

Ознакомились с языком программирования - PHP и его использованием в различных задачах, также выполнили нахождение Матрицы Достижимости и реализовали свои функции над матрицей (сложение, умножение матриц). Также научились работать с пользователем через форму ввода данных на сайте.

**Список используемой литературы**

1. Синтаксис PHP :[Электронный ресурс] URL: <http://www.php.su/learnphp/?syntax>
2. Справочник языка PHP :[Электронный ресурс] URL:<http://php.net/language.constants.syntax/>
3. Лекции Игонин Андрей Геннадьевич по интернет программированию

**Приложение №1**

<form method="post" action="dost.php">  
 Матрица достижимости<br>  
 <input name="size" type="text" placeholder="n"><br>  
 Введите матрицу nxn:<br><textarea style="resize: none" rows="20" cols="50" name="matrix">**<?=** htmlspecialchars($\_POST['matrix']) **?>**</textarea><br>  
 <input type="submit" style="width: 300px;margin: 25px" value="Do it">  
 <br>**<?php echo** "n = ", $\_POST['size']; **?>**</form>  
  
**<?php**define("SIZE", trim($\_POST['size']), **true**);  
**if** (**isset**($\_POST['matrix'])) {  
 $matrix = htmlspecialchars(trim($\_POST['matrix']));  
 $matrix = preg\_replace('# {2,}#', ' ', $matrix);  
 $matrix = preg\_replace('#(?:\r?\n){2,}#', "\r\n", $matrix);  
 $mas = explode("\r\n", $matrix);  
 **for** ($i = 0; $i < count($mas); $i++) {  
 $mas[$i] = trim($mas[$i]);  
 $elem[$i] = explode(" ", $mas[$i]);  
 }  
 **echo** "Введенная матрица M", "<br>";  
 **for** ($i = 0; $i < count($elem); $i++) {  
 **for** ($j = 0; $j < count($elem[$i]); $j++) {  
 **echo**($elem[$i][$j]);  
 **echo** " ";  
 }  
 **echo** "<br>";  
 }  
 dost($elem);  
}  
**function** dost($elem) { //ищем матрицу достижиомости: возводиим исходную смежности в степень пока степень не будет равна количеству ребер и проводим между полученными матрциами операцию дизъюнкции  
 **for** ($i = 0; $i < *SIZE*; $i++) {  
 **for** ($j = 0; $j < *SIZE*; $j++) {  
 $res[$i][$j] = 0;  
 }  
 }  
 **for** ($i = 0; $i < *SIZE*; $i++) {  
 **for** ($j = 0; $j < *SIZE*; $j++) {  
 **if** ($i == $j){  
 $new\_matrix[$i][$j] = 1;  
 } **else** {$new\_matrix[$i][$j] = 0;}  
 }  
 }  
**echo** "<br>";  
$d = 1;  
 **for** ($i = 0; $i < *SIZE*; $i++) {  
 **echo** "M^",$d;**echo** "<br>";  
 $new\_matrix = matrix\_multiply($new\_matrix, $elem);  
 //$res = $res OR $new\_matrix;  
 $res = matrix\_sum($res, $new\_matrix);  
 $d++;  
 }  
 **echo** "Результат", "<br>";  
 **for** ($i = 0; $i < *SIZE*; $i++) {  
 **for** ($j = 0; $j < *SIZE*; $j++) {  
 **echo**($res[$i][$j]);  
 **echo** " ";  
 }  
 **echo** "<br>";  
 }  
}  
**function** matrix\_multiply($new\_matrix, $elem) {  
 $product = **array**(); // Создаём массив значений  
 $row = count($new\_matrix); // Количество строк в первой матрице  
 **for** ($z=0; $z < 1; $z++) {  
 $col = count($elem[$z]); // Количество строк во втрой матрице  
 }  
 **for** ($i=0; $i < $row; $i++) {  
 **for** ($j=0; $j < $col; $j++) {  
 **for** ($k=0; $k < $col; $k++) {  
 $product[$i][$j] += $new\_matrix[$i][$k] \* $elem[$k][$j];  
 }  
 **echo** $product[$i][$j] , " ";  
 }  
 **echo** "<br>";  
 }  
 **echo** "<br>";  
 **return** $product;  
}  
  
**function** matrix\_sum($res, $new\_matrix) {  
 **for** ($i = 0; $i < count($res); $i++) {  
 **for** ($j = 0; $j < count($res[$i]); $j++) {  
 **if** ($res[$i][$j] == 1 || $new\_matrix[$i][$j] == 1){  
 $res[$i][$j] = 1;  
 } **else** {$res[$i][$j] = 0;}  
 }  
 }  
 **return** $res;  
}  
**function** validation($elem) {  
 $res = 1;  
 **for** ($i = 0; $i < count($elem); $i++) {  
 **if** (count($elem) != count($elem[$i])) {  
 $res = 'Матрица не квадратная';  
 }  
 }  
 **for** ($i = 0; $i < count($elem); $i++) {  
 **for** ($j = 0; $j < count($elem); $j++) {  
 **if** ($elem[$i][$j] == '0' || $elem[$i][$j] == '1') {  
 } **else** {  
 $res = 'Матрица не соотвествует бинарному виду';  
 **return** $res;  
 }  
 }  
 }  
 **return** $res;  
}  
**?>**