Министерство образования и науки Российской Федерации

ФГБОУ ВО Ульяновский государственный технический университет

Кафедра «Вычислительная техника»

Лабораторная работа №4

«Графы. Определение кратчайшего пути»

Выполнил студент

группы ИВТАСбд-11

Зимин Е.С.

Ульяновск, 2022

1. **Цель работы**

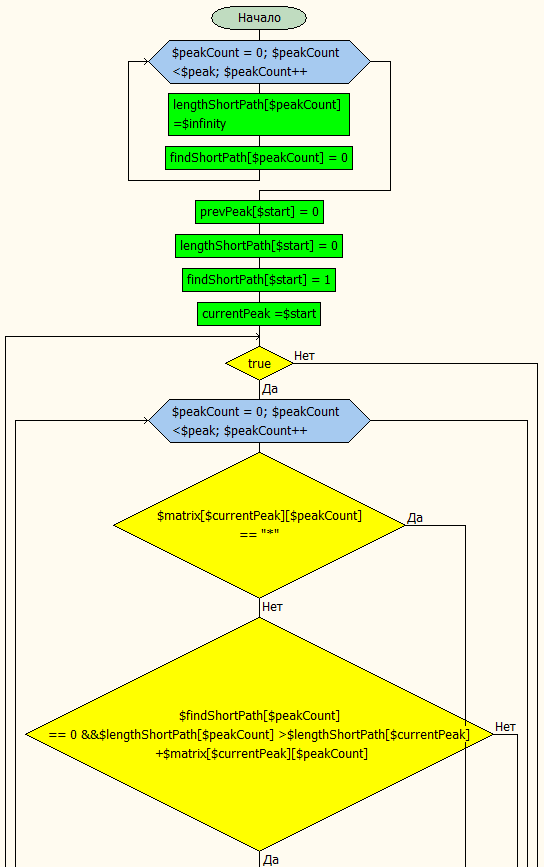
Необходимо написать программу, которая будет находить кратчайший путь в неориентированном графе. Пользователь вводит матрицу смежности графа, начальную и конечную точки.

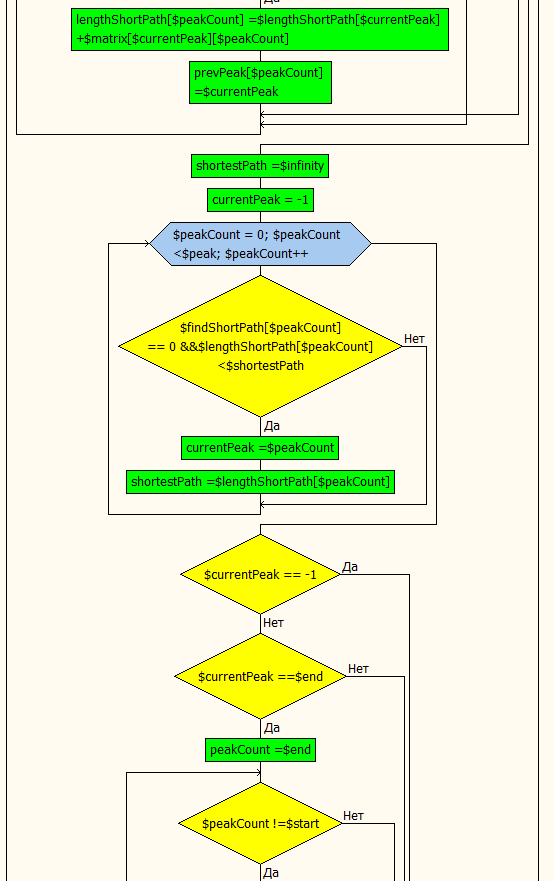
1. **Краткое описание метода выполнения**

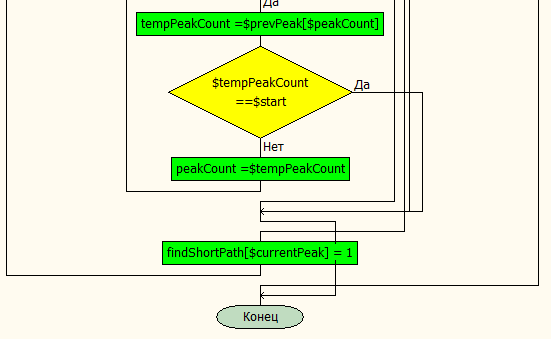
Граф - это множество упорядоченных пар G := (V, E), где V – множество вершин графа, а E – множество ребер графа. Оба множества непустые. Графы бывают ориентированные и неориентированные. Их отличие заключается в том, что в первом случае проход по ребру ориентированного графа (дуга) возможен только в одном направлении, а во втором – в обоих. Матрица смежности – квадратная матрица представления отношения вершин и ребер, где значение элемента говорит о весе ребра из i точки в j. Вес ребра — значение, поставленное в соответствие данному ребру. Кратчайший путь в графе – последовательность вершин, соединенных ребрами или дугами, у которых сумма весов этих ребер или дуг наименьшая.

Для ввода данных я использовал 3 поля и кнопку, отвечающие за ввод матрицы смежности, начальной и конечной точки и их обработку.

1. **Блок-схема алгоритма решения поставленной задачи.**







1. **Листинг кода программы**

<?php  
  
if ($\_POST['message1'] != null) {  
 $message = $\_POST['message1'];  
} else {  
 echo "Вы не ввели матрицу смежности неориентированного графа";  
 return;  
}  
  
 if ($\_POST['message2'] != null) {  
 $start = $\_POST['message2'] - 1;  
 } else {  
 echo "Вы не ввели начальную точку";  
 return;  
 }  
 if ($\_POST['message3'] != null) {  
 $end = $\_POST['message3'] - 1;  
 } else {  
 echo "Вы не ввели конечную точку";  
 return;  
 }  
  
  
  
$matrixElements = preg\_split('/[ \n]/', $message);  
$sizeMatrix = sqrt(count($matrixElements));  
  
if (floor($sizeMatrix) != $sizeMatrix) {  
 echo "У вас не квадратная матрица смежности";  
 return;  
}  
  
$matrix = array($sizeMatrix);  
  
for($x = 0; $x < $sizeMatrix; $x++) {  
 $matrix[$x] = array($sizeMatrix);  
 for($y = 0; $y < $sizeMatrix; $y++)  
 $matrix[$x][$y] = $matrixElements[$x + $y \* $sizeMatrix];  
}  
  
  
 $flag = true;   
 for ($x = 0; $x < $sizeMatrix; $x++) {  
 if ($matrix[$x][$x] != 0) {  
 if ($flag)  
 echo "Это матрица смежности ориентированного графа :(";  
  
 echo "<br> <br>Ошибка в элементе с индексом: ";  
 echo $x + 1;  
 echo " ";  
 echo $x + 1;  
 echo "<br>Этот индекс должен быть равен '\*' <br>";  
 $flag = false;  
 }  
 for ($y = 0; $y < $sizeMatrix; $y++) {  
 if ($matrix[$x][$y] != $matrix[$y][$x]) {  
 if ($flag) {  
 echo "Это матрица смежности ориентированного графа :(<br>";  
 $flag = false;  
 }  
 echo "<br>Возможно ошибка в элементе с индексом: ";  
 echo $x + 1;  
 echo " ";  
 echo $y + 1;  
 }  
 }  
 }  
  
 if (!$flag)  
 return;  
  
  
 $infinity = 9999;   
 $peak = sqrt(count($matrixElements));   
  
 $findShortPath = array(count($matrixElements));  
 $lengthShortPath = array(count($matrixElements));  
 $prevPeak = array(count($matrixElements));  
 for ($peakCount = 0; $peakCount < $peak; $peakCount++)

{  
 $lengthShortPath[$peakCount] = $infinity;  
 $findShortPath[$peakCount] = 0;  
 }  
  
 $prevPeak[$start] = 0;

$lengthShortPath[$start] = 0;

$findShortPath[$start] = 1;   
 $currentPeak = $start;   
  
 while (true) {

for ($peakCount = 0; $peakCount < $peak; $peakCount++) {  
 if ($matrix[$currentPeak][$peakCount] == "\*")  
 continue;  
  
 if ($findShortPath[$peakCount] == 0 && $lengthShortPath[$peakCount] > $lengthShortPath[$currentPeak] + $matrix[$currentPeak][$peakCount])   
 {  
 $lengthShortPath[$peakCount] = $lengthShortPath[$currentPeak] + $matrix[$currentPeak][$peakCount];  
 $prevPeak[$peakCount] = $currentPeak;  
 }  
 }  
 $shortestPath = $infinity;  
 $currentPeak = -1;  
  
 for ($peakCount = 0; $peakCount < $peak; $peakCount++)   
 {  
 if ($findShortPath[$peakCount] == 0 && $lengthShortPath[$peakCount] < $shortestPath)  
 {  
 $currentPeak = $peakCount;  
 $shortestPath = $lengthShortPath[$peakCount];  
 }  
 }  
 if ($currentPeak == -1) {  
 echo "Нет пути из вершины ";  
 echo $start + 1;  
 echo " в вершину ";  
 echo $end + 1;  
 break;  
 }  
 if ($currentPeak == $end)  
 {   
 echo "Кратчайший путь из вершины ";  
 echo $start + 1;  
 echo " в вершину ";  
 echo $end + 1;  
 echo ":<br>";  
 $peakCount = $end;  
  
 while ($peakCount != $start) {  
 $tempPeakCount = $prevPeak[$peakCount];  
 echo $peakCount + 1;  
 if ($tempPeakCount == $start)  
 break;  
 echo " ← ";  
 $peakCount = $tempPeakCount;  
 }  
  
 echo " ← ";  
 echo $start + 1;  
 echo "<br>";  
 echo "Длина пути: ";  
 echo $lengthShortPath[$end];  
 break;  
 }  
 $findShortPath[$currentPeak] = 1;  
 }  
 echo "<br><br><br><br><br><br><br>";  
}

1. **Выводы**

Программа работает корректно. Верно определяет пути от начальной до конечной точки в графе и сумму весов.

1. **Список литературы**
2. <https://www.php.net/manual/ru/langref.php> – справочник по PHP
3. <https://php.ru/manual/> – методичка PHP