Министерство образования и науки Российской Федерации ФБГОУ ВО Ульяновский государственный технически университет

Кафедра «вычислительная техника»

**Лабораторная работа №4**

«Нахождение кратчайшего пути в графе»

Выполнила студентка

группы ИВТАСбд-11

Денисова Д.В.

Ульяновск, 2022

**Оглавление.**

1. Цель работы…………………………………………………….…………..2
2. Описание метода выполнения………………………………...…………..2
3. Блок-схема алгоритма………………………………………………….......2
4. Вывод………………………………………………………………………..8
5. Список использованной литературы……...………………………………8
6. **Цель работы.**

Необходимо написать программу нахождения кратчайшего пути в графе. Формат ввода: матрица смежности, начальная вершина, конечная вершина. При выводе необходимо показывать весь маршрут, путем перечисления вершин и суммарная стоимость пути.

1. **Описание метода выполнения.**

**Матрица смежности —** один из способов представления графа в виде матрицы**.** Для того, чтобы пользователь мог вводить данные необходимо написать небольшую html-структуру, состоящую из трех полей для ввода, полей для вывода и кнопку, по которой произойдут операции над множествами. Код проводимых операций будет написан на JavaScript. Мы получаем наши массивы в html-структуру на PHP и по id передаём в JavaScript.

1. **Блок-схема алгоритма.**
2. Для проверки на корректность ввода данных пользователем, я пробегаюсь по нашим массивам с определенными условиями и если ввод элементов массива некорректен, то у пользователя высвечивается предупреждение, в котором написано в каком массиве и в каком элементе был допущена ошибка ввода.

function checkmass(str){

mass='fal';

if(str.length){

mass=str.split("\n");

for(let i=0; i<mass.length; i++){

for(let j=0; j<mass[i].length; j++) {

if ((mass[i][j]>'9' || mass[i][j]<'0') && mass[i][j]!=='#' || mass.length!==mass[i].length || mass[i][j]!==mass[j][i]) {

error\_text = 'Ошибка при вводе: '+ '\n' + str + '\n' +' в элементе: ' + mass[i][j];

mass = 'fal';

break;

}

}

}

}

else {

error\_text="Вы не ввели массив"

}

return mass;

}

1. В этой функции я использую алгоритм Дейкстры. Этот алгоритм ищет кротчайший путь от одной вершины до всех, поэтому я решила использовать именно его, так как он похож на поставленную задачу, но искала не до всех вершин, а до одной определённой вершины.

function shortest\_way(mass, s\_vertex, f\_vertex){

let vertex;

let vis2=" ";

let infinity=1000; // Бесконечность

let p = mass[0].length; // Количество вершин в графе

let arr=mass; // Матрица смежности графа

let start=parseInt(s\_vertex); // Номер исходной вершины

let final = parseInt(f\_vertex);// Номер конечной вершины

let binary=[mass[0].length]; //Массив, содержащий единицы и нули для каждой вершины,

// x[i]=0 - еще не найден кратчайший путь в i-ю вершину,

// x[i]=1 - кратчайший путь в i-ю вершину уже найден

let t=[mass[0].length];

let h=[];

let u; // Счетчик вершин

for (u=0;u<p;u++) {

t[u]=infinity;

binary[u]=0;

}

h[start]=0;

t[start]=0;

binary[start]=1;

vertex=start;

while(1) {

for (u = 0; u < p; u++) {

if (parseInt(arr[vertex][u]) === 0) {

continue;

}

if (binary[u] === 0 && t[u] > parseInt(t[vertex]) + parseInt(arr[vertex][u]))

{

t[u] = parseInt(t[vertex]) + parseInt(arr[vertex][u]);

h[u] = vertex;

}

}

let w = infinity;

vertex = -1;

for (u = 0; u < p; u++)

{

if (parseInt(binary[u]) === 0 && parseInt(t[u]) < w) {

vertex = u;

w = parseInt(t[u]);

}

}

if(vertex===-1)

{

vis="Нет пути из вершины "+start+" в вершину "+final+".";

break;

}

if(vertex===final) {

vis="Кратчайший путь из вершины "+start+" в вершину "+final+":" + '\n';

u=final;

while(u!==start)

{

vis2+='->'+u+' ';

u=parseInt(h[u]);

}

vis2+=" "+start;

break;

}

binary[vertex]=1;

}

let res=vis2.split(" ").reverse().toString();

let res\_out=vis;

for(let j =0; j<res.length; j++){

if(res[j]!==","){

res\_out+=res[j];

}

}

res\_out+=". Вес пути: "+t[final]+'.';

return res\_out;

}

3. В этой функции мы сначала вызываем функцию для проверки на корректность ввода массивов, если он выведен некорректно, то выводим на экран пользователю нашу переменную, в которой хранится информация об ошибке. Затем вызываем нашу функцию, собираем весь результат операций в одну строку и выводим ее.

function calculation\_l4() {

let result="";

var m\_matrix=document.getElementById('mass1')

var start=document.getElementById('mass2')

var final=document.getElementById('mass3')

if((mass\_matrix=checkmass(m\_matrix.value))=='fal'){

alert(error\_text);

}

if((final\_vertex=checkmass(final.value))=='fal'){

alert(error\_text);

}

if((starting\_vertex=checkmass(start.value))=='fal'){

alert(error\_text);

}

if((mass\_matrix=checkmass(m\_matrix.value))!='fal' && (final\_vertex=checkmass(final.value))!='fal' && (starting\_vertex=checkmass(start.value))!='fal'){

result = 'Результат:'+ '\n' + shortest\_way(mass\_matrix, start.value, final.value);

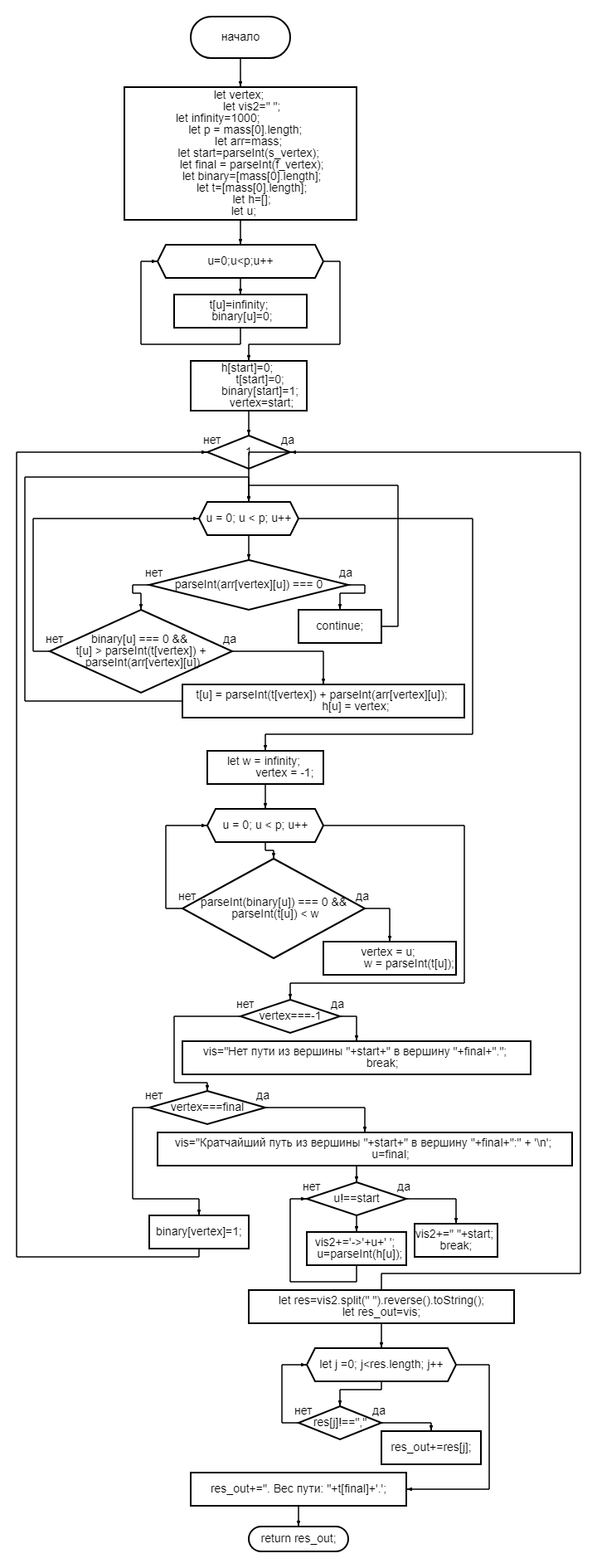
}

if(result!=""){

document.getElementById('outresult\_lab4').innerText = result;

}

4.Блок-схема функции shortest\_way()-нахождение кротчайшего пути



1. **Вывод.**

Программа протестирована, работает успешно. Она определяет кратчайший путь в графе.

1. **Список использованной литературы.**

<https://habr.com/en/post/111361/> - Алгоритм Дейкстры.

<https://learn.javascript.ru/> - Современный учебник JavaScript