Минестерство образования и науки Российской Федерации

ФГБОУ ВО Ульяновский государственный технический университет

Кафедра «вычислительная техника»

Лабораторная работа №2

«Операции над множествами»

Выполнил студент

Группы ИВТАСбд-11

Долгов А. П.

Ульяновск, 2022

**Цель работы:**

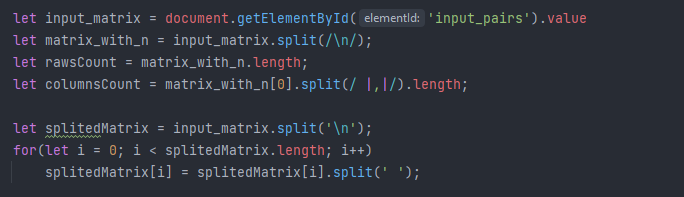
Разработать программу, определяющую свойства отношения. Вводить нужно матрицу смежности. Необходимо выполнить проверку корректности ввода отношения. Определяемые свойства: рефлексивность, симметричность, кососимметричность, транзитивность

**Описание метода выполнения.**

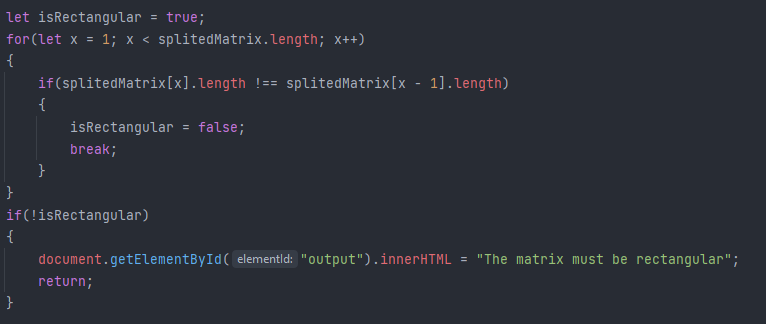
Бинарным отношением между элементами множества и называется любое подмножество множества , то есть . Отношение рефлексивно, если для любых х из множества А существуют отношение xRx. Отношение симметрично, если для любых х и y из множества А существует xRy, из которого следует yRx. Отношение кососимметрично, если для любых х и у из множества А существует xRy и yRx, из которых следует, что x=y. Отношение транзитивно, если для любой тройки элементов x, y, z из А и таких бинарных отношений, как xRy и yRx следует xRz.

Для того, чтобы пользователь мог вводить матрицу на сайте создана специальное поле textarea. Чтобы программа выполнилась, нужно нажать на кнопку “Press”, после чего будет выведены соответствующие свойства матрицы. Если же данные введены некорректно, то вместо свойств выведется текст, указывающий на ошибки. Код программы написан на JavaScript. Кроме того, для улучшения визуального оформления применяется CSS стили.

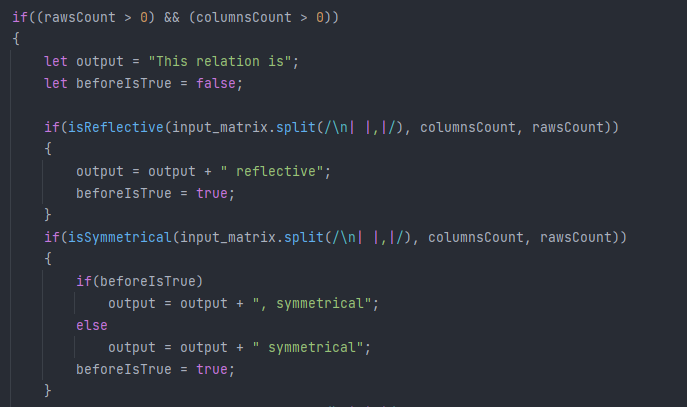
По нажатию на кнопку, вызывается js код. Сперва он переводит строку в двумерный массив матрицы смежности:

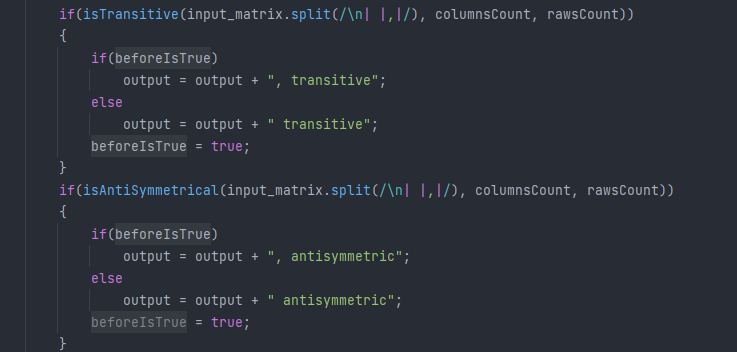


Затем идут некоторые проверки правильности формата ввода матрицы:



Дальше, с помощью нескольких условий определяется свойства матрицы и добавляются соответствующие строки к выходной строке:





В конце происходит вывод строки:

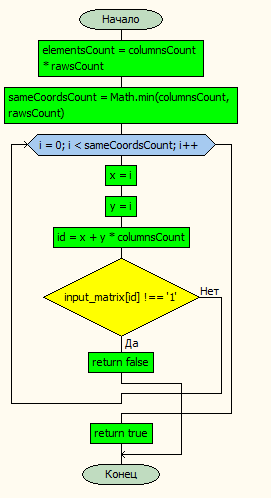


Пройдёмся по алгоритмам определения свойств матрицы.

*Рефлективность*

Начнём с алгоритма, проверяющего свойство рефлективности.

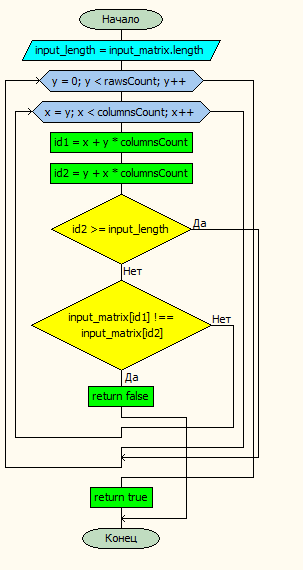
function isReflective(input\_matrix, columnsCount, rawsCount)  
{  
 let elementsCount = columnsCount \* rawsCount;  
 let sameCoordsCount = Math.min(columnsCount, rawsCount);  
  
 for(let i = 0; i < sameCoordsCount; i++)  
 {  
 let x = i;  
 let y = i;  
  
 let id = x + y \* columnsCount;  
 if(input\_matrix[id] !== '1')  
 return false;  
 }  
 return true;  
}



Всё, что делает этот алгоритм, это проверяет, во всех ли элементах на диагонали с левого верхнего до правого нижнего угла находится единицы. Если это так, то это означает, что отношения между одинаковыми элементами всегда существует, на графе это выглядит как петли у каждой вершины.

*Симметричность*

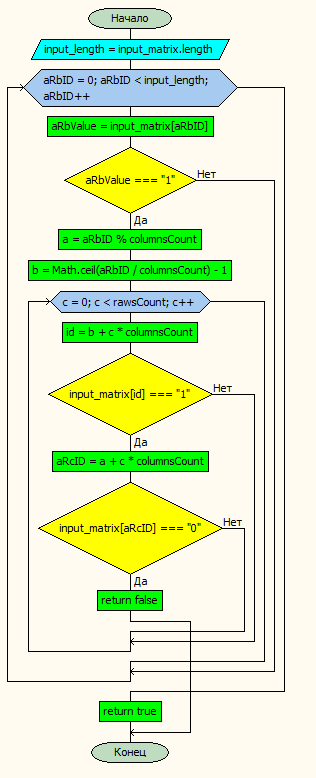
function isSymmetrical(input\_matrix, columnsCount, rawsCount)  
{  
 let input\_length = input\_matrix.length;  
 for(let y = 0; y < rawsCount; y++)  
 {  
 for(let x = y; x < columnsCount; x++)  
 {  
 let id1 = x + y \* columnsCount;  
 let id2 = y + x \* columnsCount;  
 if(id2 >= input\_length)  
 break;  
 if(input\_matrix[id1] !== input\_matrix[id2])  
 return false;  
 }  
 }  
 return true;  
}



Алгоритм проверяет, все ли элементы, симметричны относительно главной. Если это так, то это означает, что матрица симметрична.

*Транзитивность*

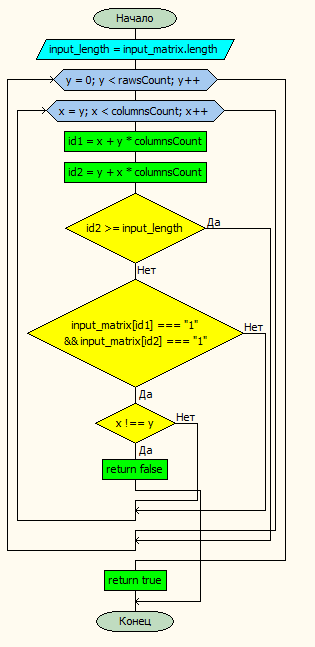
function isTransitive(input\_matrix, columnsCount, rawsCount)  
{  
 let input\_length = input\_matrix.length;  
 for(let aRbID = 0; aRbID < input\_length; aRbID++)  
 {  
 let aRbValue = input\_matrix[aRbID];  
 if(aRbValue === "1")  
 {  
 let a = aRbID % columnsCount;  
 let b = Math.ceil(aRbID / columnsCount) - 1;  
  
 for(let c = 0; c < rawsCount; c++)  
 {  
 let id = b + c \* columnsCount;  
 if(input\_matrix[id] === "1")  
 {  
 let aRcID = a + c \* columnsCount;  
 if(input\_matrix[aRcID] === "0")  
 return false;  
 }  
 }  
 }  
 }  
 return true;  
}



Алгоритм проверки транзитивности находит все возможные комбинации трёх элементов, и проверяет, является ли отношения между ними транзитивными. Если они транзитивны, то это означает, что если есть отношение между элементом и элементом , а также есть отношение между элементом и элементом , то обязательно должно быть отношение между элементом и элементом .

*Кососимметричность (антисимметричность)*

function isAntiSymmetrical(input\_matrix, columnsCount, rawsCount)  
{  
 let input\_length = input\_matrix.length;  
 for(let y = 0; y < rawsCount; y++)  
 {  
 for(let x = y; x < columnsCount; x++)  
 {  
 let id1 = x + y \* columnsCount;  
 let id2 = y + x \* columnsCount;  
 if(id2 >= input\_length)  
 break;  
 if(input\_matrix[id1] === "1" && input\_matrix[id2] === "1")  
 {  
 if(x !== y)  
 return false;  
 }  
 }  
 }  
 return true;  
}



При кососимметричности в отношении не должно быть симметричных элементов, кроме тех элементов, которые находятся на главной диагонали, то есть тех, которые имеют отношения с самим собой. Алгоритм проверяет отсутствие симметрии относительно главной диагонали, и если это оказывается так, то функция возвращает истину.

**Вывод**

Данная программа протестирована и успешно работает. Программа позволяет ввести матрицу отношения, и получить все свойства, которыми обладает данное отношение. При вводе некорректных данных программа будет сообщать об этом.

**Список источников**

1. ДМ. Бинарные отношения, часть 1. 22 сентября 2020 года. URL:<https://www.youtube.com/watch?v=8REh96AgsW4&t=1237s>
2. Википедия. Отношения (теория множеств). URL: <https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BD%D0%BE%D1%88%D0%B5%D0%BD%D0%B8%D0%B5_(%D1%82%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2)>