МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

**«Дальневосточный федеральный университет»**

|  |
| --- |
|  |

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**Кафедра информационной безопасности**

**Интеллектуальные компьютерные системы**

**ОТЧЕТ**по лабораторной работе №5

на тему «**Алгоритмы распознавания**»

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Выполнил студент гр. Б8116-09.03.01 | | |  | Гусев М.Д. | |  | | | Проверил | | |  | Ю.С. Москаленко | | дата | | |  | | | зачтено/не зачтено | | |

г. Владивосток

2019

**Постановка задачи и входные данные**

В качестве входных данных выступает матрица заданной размерности, разделенная на заданное количество групп, заполненная чем угодно, а также строка, также заполненная чем-угодно

Задача состоит в том, чтобы определить к какой группе входных данных относится введенная строка.

**Реализация**

Изначально пользователю предоставляются поля, в которых можно задать столбцов, количество столбцов и количество строк в каждой группе матрицы матрицы. Это было реализовано с помощью метода promt(,), который выводит пользователю диалоговое окно с заданным текстом, а также поле для ввода данных типа integer.

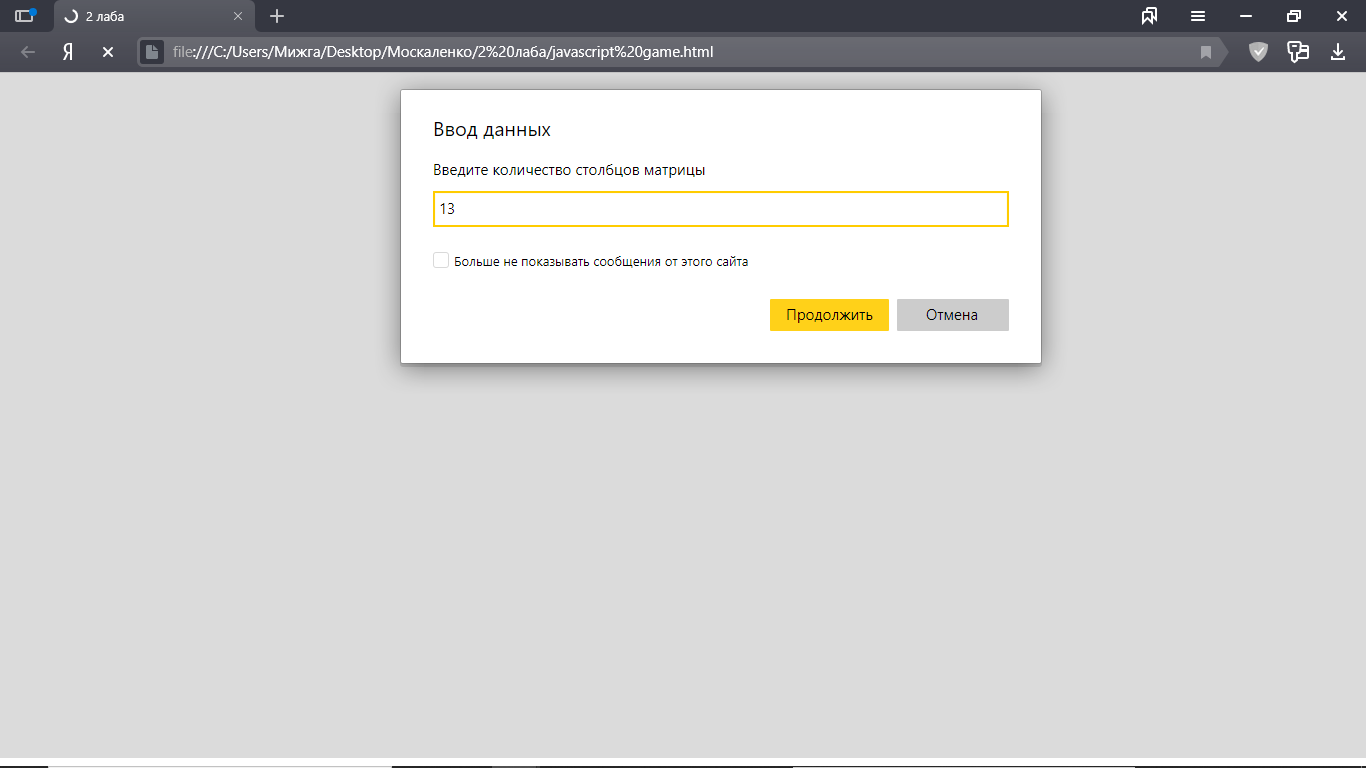
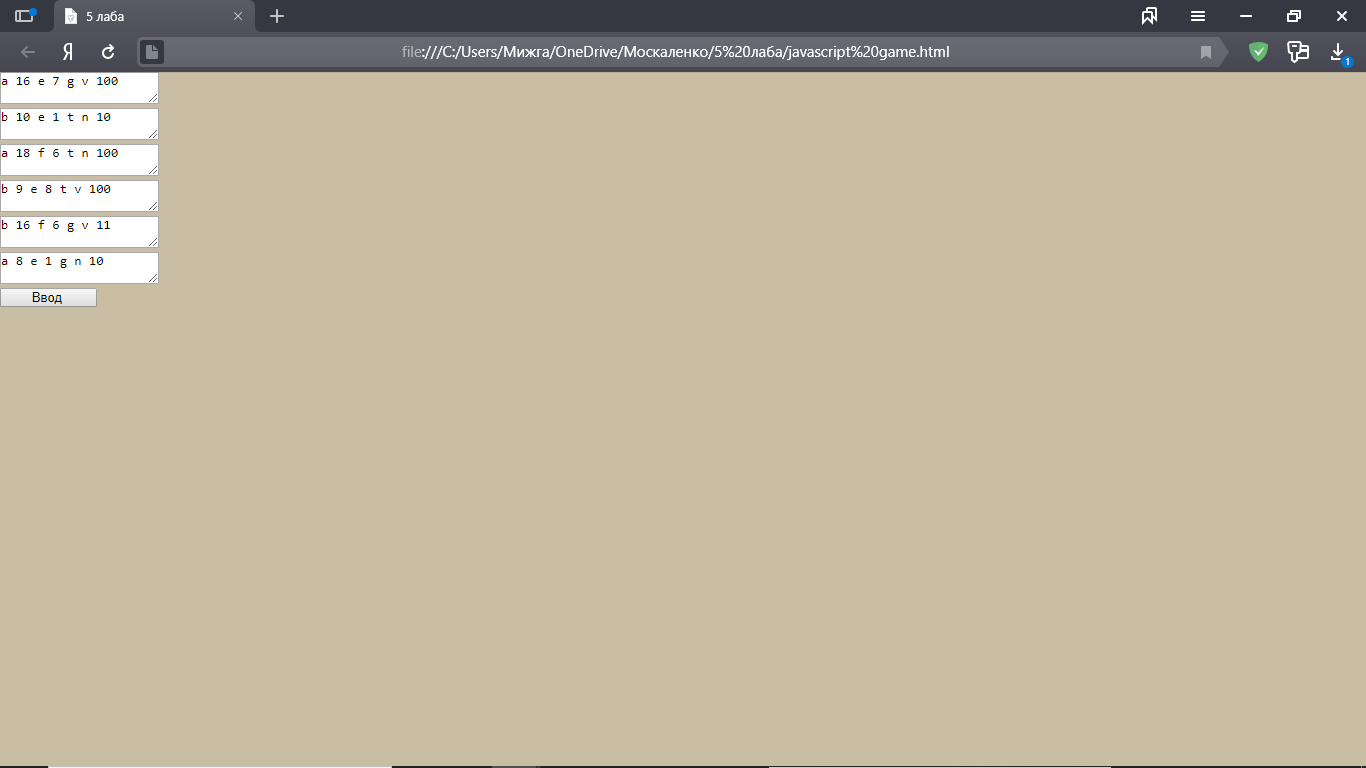


рис. 1 – диалоговое окно для ввода данных.

Далее пользователю выводятся другие поля для ввода данных, представляющие собой строки, которые можно полностью заполнять аналогичными строками, которые будут в матрице. После нажатия на кнопку «Ввод» выведется еще одна форма для ввода строки, принадлежность которой и нужно будет определить к той или иной группе.

  
Рис.2 – построчный ввод данных матрицы

После ввода данных пользователю отображается заданная матрица. Для этого введенные данные были сохранены в двумерном массиве.   
Также создается еще один двумерный массив, каждый элемент которого будет являться блоком div, которому задается нужный визуальный стиль. Именно внутрь блоков div помещены данные из первого массива.  
Если кратко, в первом массиве хранятся введенные цифры, во втором – блоки div, которые являются визуальной оболочкой для цифр.

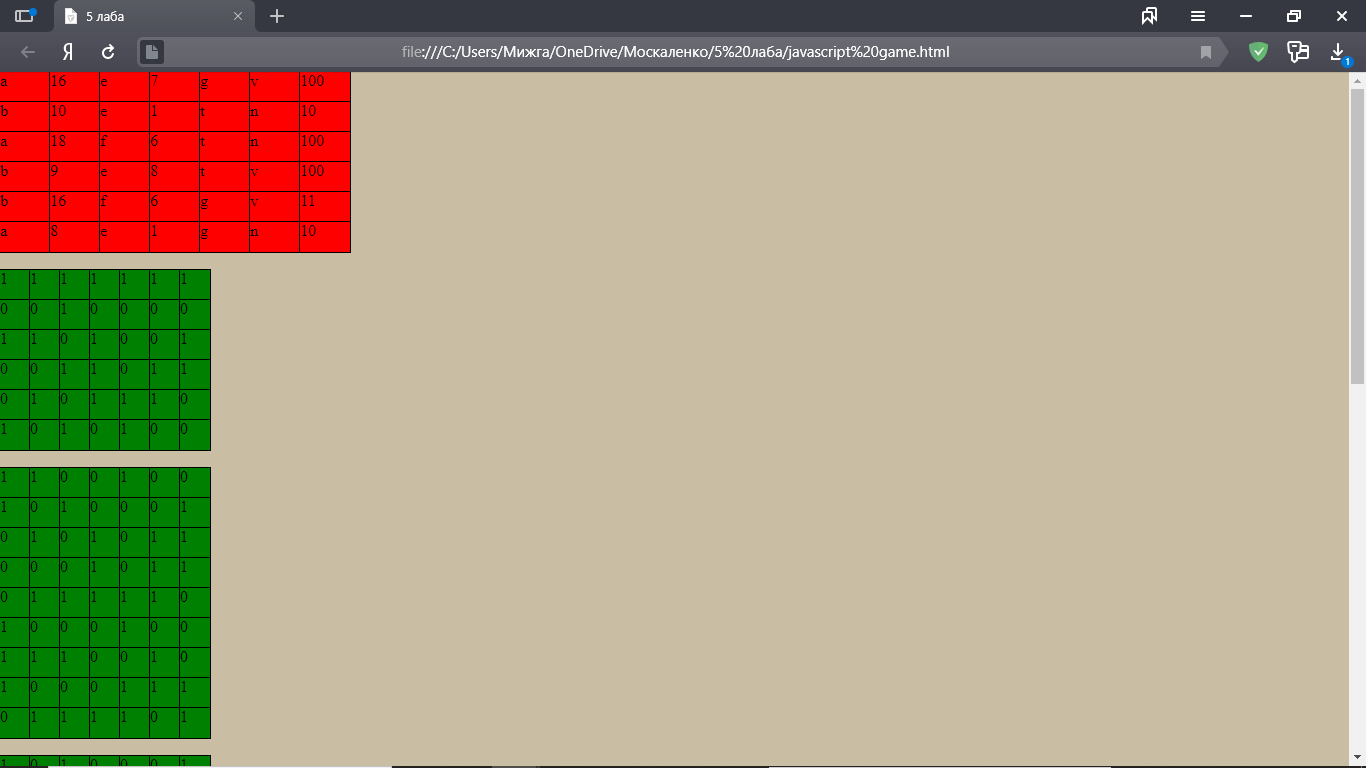


рис. 3 – матрица введенных элементов

Затем выводится матрица, состоящая из единиц и нулей. Получена она таким образом:  
Сначала вычисляется первая строка матрицы. Если в введенной матрице в первый элемент столбца равен букве(слову), то она автоматически становится единицей, а все, отличающиеся от данной буквы(слова), символы будут в столбце нулями.   
Если же первый символ в столбце является цифрой, то вычисляется среднее значение в столбце, и все числа больше этого значения будут единицами, а меньше – нулями.

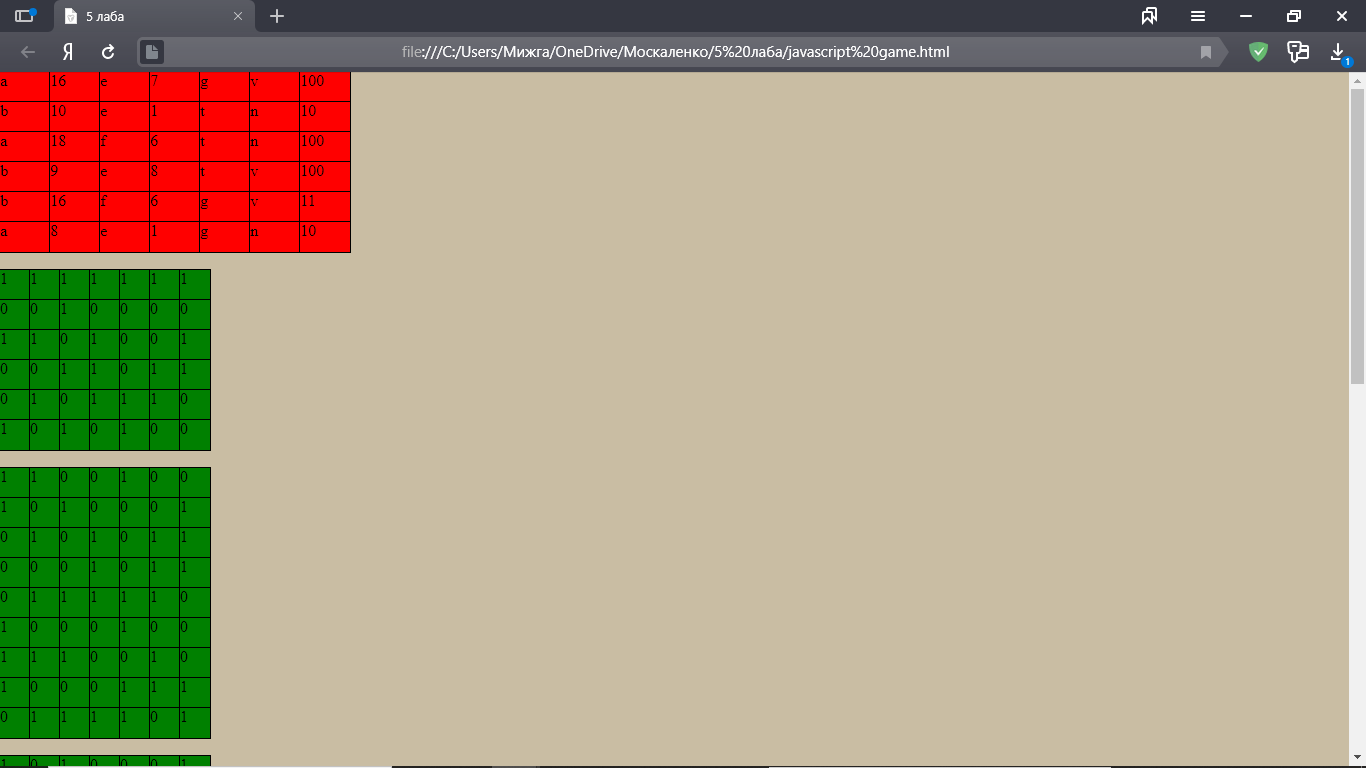


рис.4 – матрица введенных элементов и матрица булевских значений

Затем строится элементарная предикатная матрица. Сравниваются строки двух групп. Если значения совпадают, то выводится ноль, в противном случае – единица.

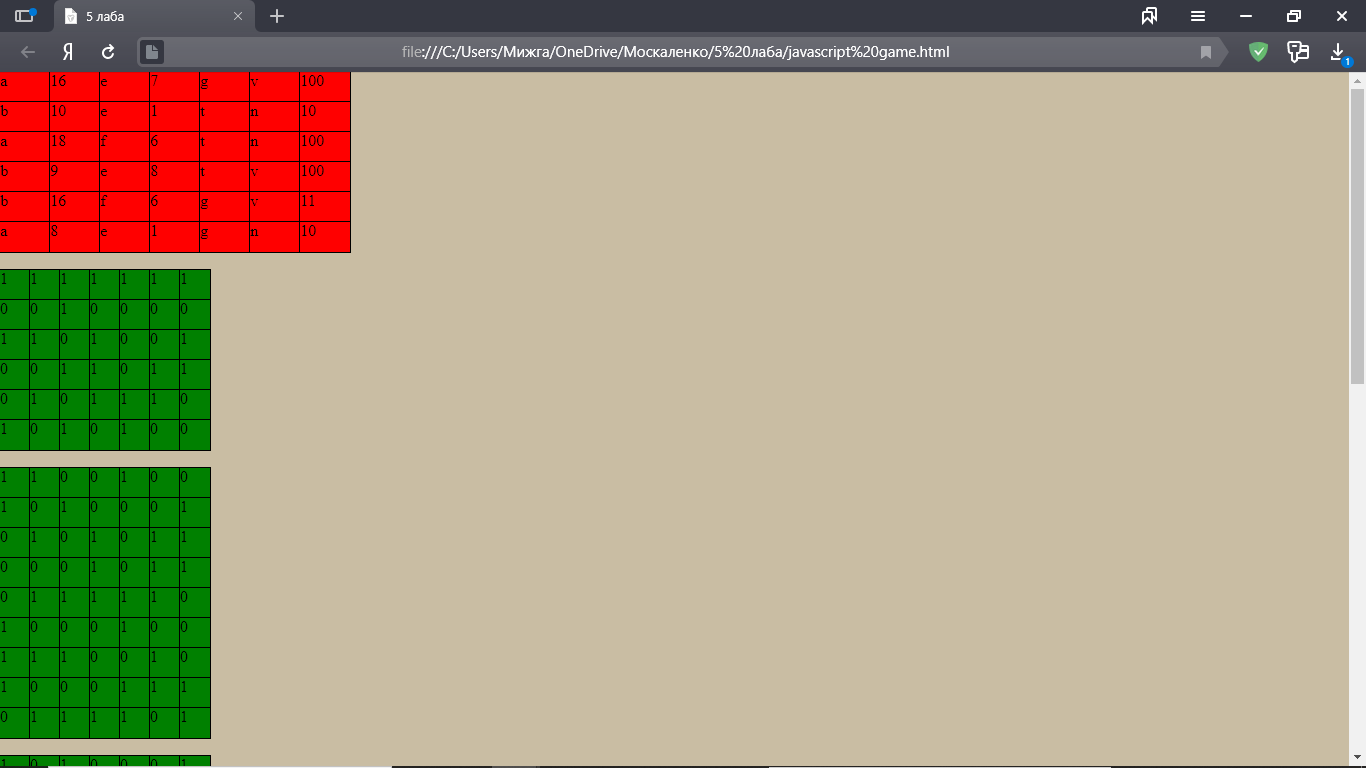


рис. 5 – элементарная предикатная матрица

После вывода элементарной предикатной матрицы идет ее минимизирование. Программа ищет строки с минимальным количеством единиц. Если другие строки имеют единицы на тех же позициях, то они удаляются.

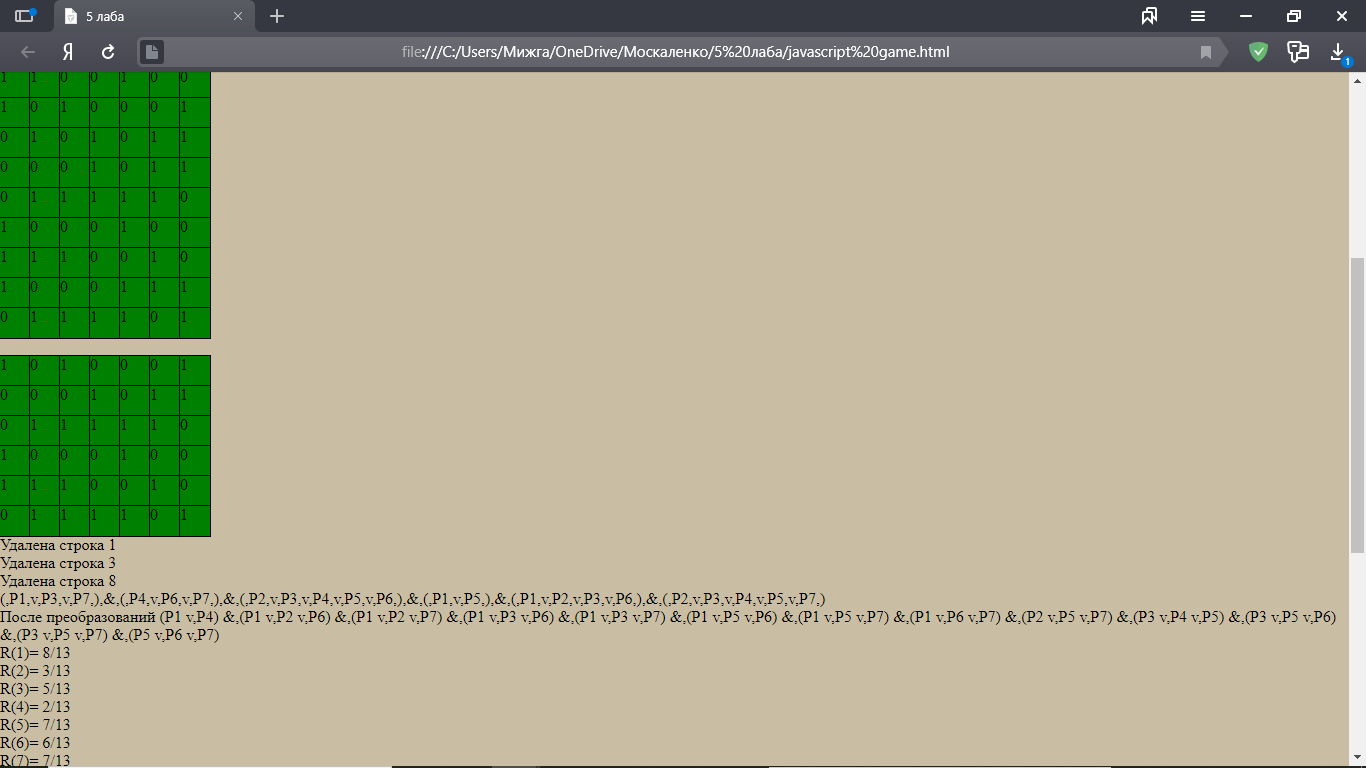


Рис.6 – минимизированная элементарная предикатная матрица.

Далее на основе полученной матрицы выстраивается множество. В строках ищутся единицы, позиция каждой единицы – индекс P в множестве. Каждая новая строка – новое подмножество.

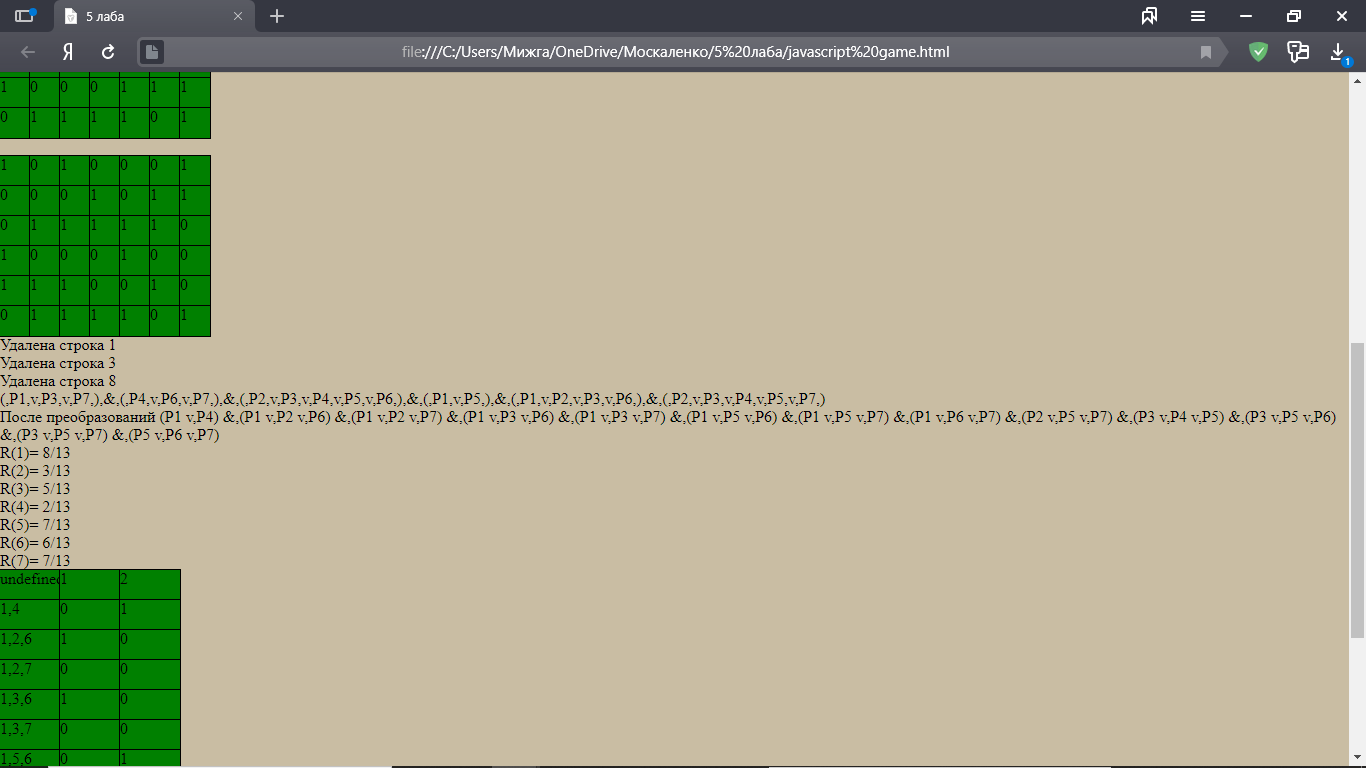


Рис.7 – минимизированная элементарная предикатная матрица и множество на ее основе.

После этого полученное множество вычисляется и минимизируется. В итоговом множестве подсчитывается количество каждого элемента в подмножествах и количество самих подмножеств. Итог также выводится на экран.

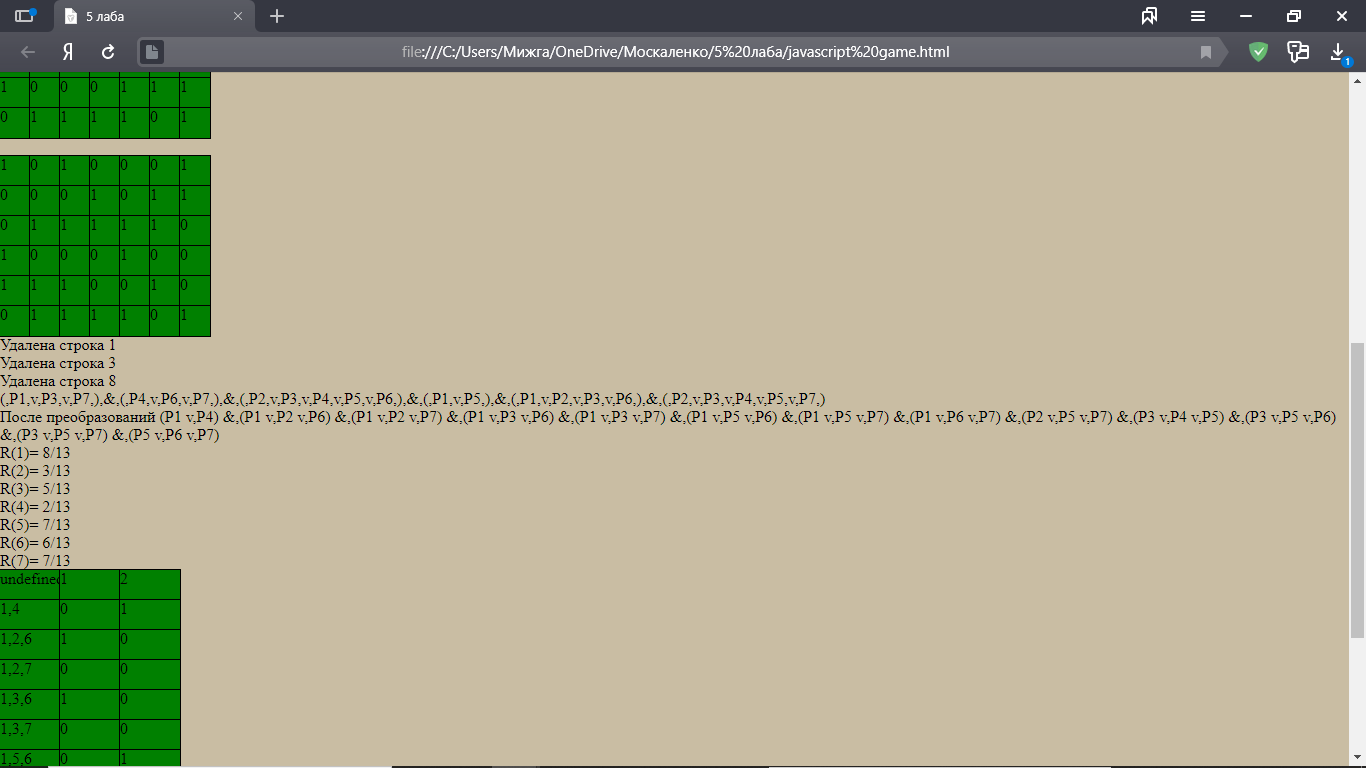


Рис.7 – начальное и итоговое множество, количество каждого элемента в подмножествах

В конце выводится таблица, в которой по строкам выводятся получившиеся подмножества, а в столбцах количество групп. Здесь идет подсчет совпадений введенной строки с матрицей по индексам, извлеченным из итоговых подмножеств. А также итоговый счет, благодаря которому можно определить к какой группе относится введенная строка.

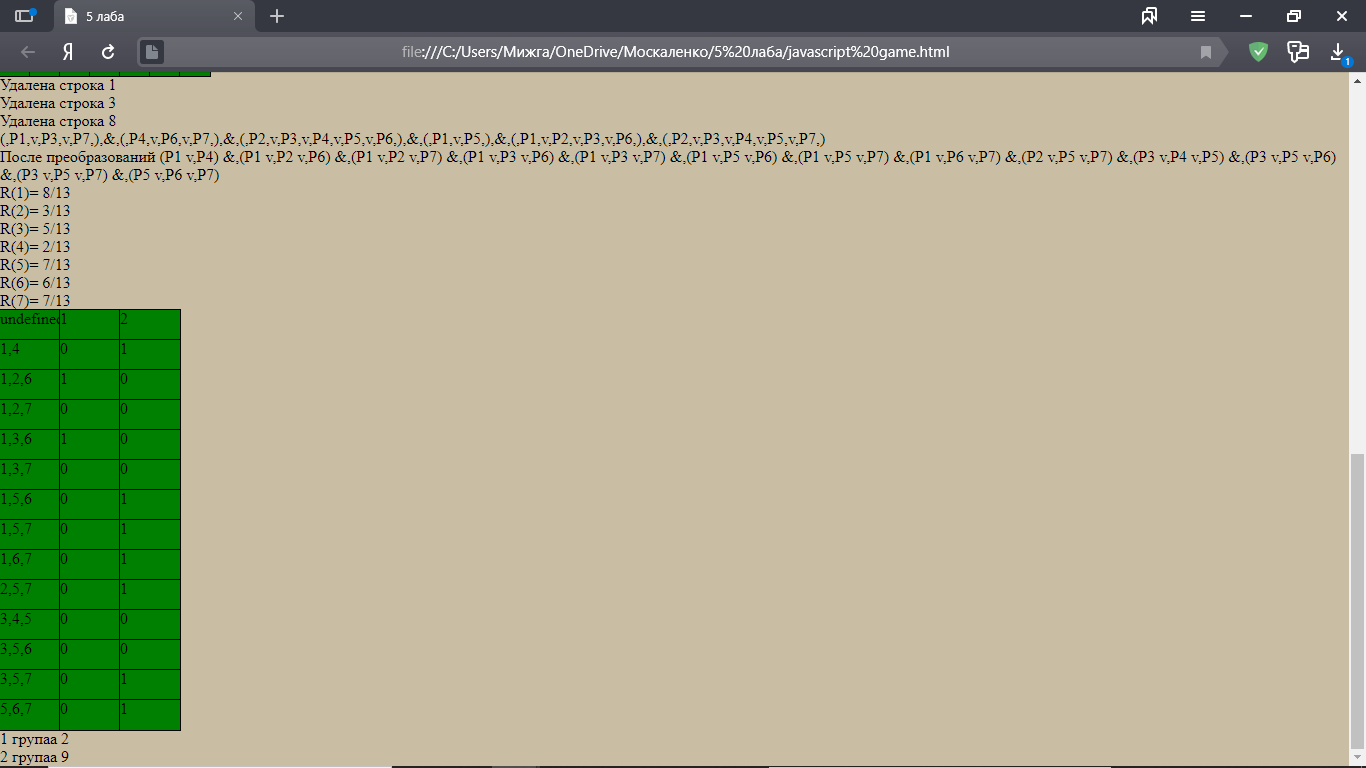


Рис.8 – итоговая таблица совпадений строки с булевской матрицей и итоговый счет.

Код программы

var count=0; count1 = 0;

var command,command1;

var mas = [];mas1 = []; mas2 = []; mas3 = [];variable = []; c\_var =[];apm = []; c\_apm =[]; ed =[];id=[]; del = [];copy = [];dd = [];ddd=[];d=[];

var block,val1,val2,count3;

var val4=0;

var val3 =[];var srav =0;

window.onload = function()

{

block = document.createElement('div');

document.body.appendChild(block);

val1 = parseInt(prompt("Введите количество групп", 2));

val2 = parseInt(prompt("Введите количество столбцов", 2));

for(var i=0;i< val1;i++){

val3[i] = parseInt(prompt("Введите количество строк в группе "+(i+1), 2));

}

for(var i=0;i<val3.length;i++){

val4 += val3[i];

}

for(var i=0;i<val4;i++){

text1=document.createElement('textarea');

document.body.appendChild(text1);

br=document.createElement('br');

br.id = 'i'+i;

document.body.appendChild(br);

text1.id=i;

}

for(var i=0;i<1;i++){

input = document.createElement('input');

document.body.appendChild(input);

input.type = 'button';

input.value=" Ввод ";

}

br1 = document.createElement('br');

document.body.appendChild(br1);

input.onclick=function(){

count =0;

for(var i=0;i<val4;i++){

mas1[i] = [];

count3=0;

t=document.getElementById(i).value;

mas1[i] = t.split(" ");

document.getElementById(i).style.display = 'none';

input.style.display = 'none';

br1.style.display = 'none';

document.getElementById('i'+i).style.display = 'none';

}

text1=document.createElement('textarea');

document.body.appendChild(text1);

br=document.createElement('br');

document.body.appendChild(br);

for(var i=0;i<1;i++){

input = document.createElement('input');

document.body.appendChild(input);

input.type = 'button';

input.value=" Ввод ";

}

input.onclick=function(){

t=text1.value;

dd = t.split(" ");

text1.style.display = 'none';

br.style.display = 'none';

input.style.display = 'none';

matrix();

}

}

}

function matrix()

{

for(var i = 0;i<val4;i++)

{

mas[i] = [];

mas3[i] = [];

for (var j = 0;j<val2;j++)

{

div = document.createElement('div');

document.body.appendChild(div);

mas[i][j] = div;

mas[i][j].style.width='50px';

mas[i][j].style.height='30px';

mas[i][j].style.background='red';

mas[i][j].style.outline = '1px solid #000';

mas[i][j].style.display = 'inline-block';

mas[i][j].innerHTML = mas1[i][j];

}

br1 = document.createElement('br');

document.body.appendChild(br1);

}

for(var i=0;i<val2;i++){

var check = mas1[0][i]\*0;

if(isNaN(check) === false)variable[i]=0;

else variable[i]=1;

}

for(var j=0;j<val2;j++){

if(variable[j] == 0){

var medium = 0;

for(var k=0;k<val4;k++){

medium += mas1[k][j]\*1;

}

c\_var[j] = medium/val4;

}

else if(variable[j] == 1){

c\_var[j] = mas1[0][j];

}

}

for(var i=0;i<val2;i++){

if(variable[i]==0){

for(var j=0;j<val4;j++){

if(c\_var[i] <= mas1[j][i]){

mas3[j][i] = 1;

}

else{

mas3[j][i] = 0;

}

}

}

else if(variable[i]==1){

for(var j=0;j<val4;j++){

if(c\_var[i] == mas1[j][i]){

mas3[j][i] = 1;

}

else{

mas3[j][i] = 0;

}

}

}

}

for(var i=0;i<val2;i++){

if(variable[i]==0){

if(c\_var[i] <= dd[i]){

d[i] = 1;

}

else{

d[i] = 0;

}

}

else if(variable[i]==1){

if(c\_var[i] == dd[i]){

d[i] = 1;

}

else{

d[i] = 0;

}

}

}

count = 0;

br1 = document.createElement('br');

document.body.appendChild(br1);

for(var i=0; i<val4;i++)

{

mas2[i] = [];

for(var j=0;j<val2;j++)

{

div = document.createElement('div');

document.body.appendChild(div);

mas2[i][j] = div;

mas2[i][j].style.width='30px';

mas2[i][j].style.height='30px';

mas2[i][j].style.background='green';

mas2[i][j].style.outline = '1px solid #000';

mas2[i][j].style.display = 'inline-block';

mas2[i][j].innerHTML = mas3[i][j];

}

br1 = document.createElement('br');

document.body.appendChild(br1);

}

count = 0;

for(var i=0;i<val1;i++){

for(var j=i+1;j<val1;j++){

srav += val3[i]\*val3[j];

}

}

br1 = document.createElement('br');

document.body.appendChild(br1);

mas2.splice(0,mas2.length);

var sch\_apm = 0;

count = val3[sch\_apm];

while(sch\_apm != val1-1 ){

for(var i=0;i<val4;i++){

for(var j=i+count;j<val4;j++){

for(var k=0;k<val2;k++){

if(mas3[i][k] != mas3[j][k]){

apm.push(1);

}

else{

apm.push(0);

}

}

}

count -=1;

var x=0;

if(count == 0){

sch\_apm++;

if(sch\_apm == val1 - 1)break;

count = val3[sch\_apm];

}

}

}

for(var i=0; i<srav;i++)

{

mas2[i] = [];

for(var j=0;j<val2;j++)

{

div = document.createElement('div');

document.body.appendChild(div);

mas2[i][j] = div;

mas2[i][j].style.width='30px';

mas2[i][j].style.height='30px';

mas2[i][j].style.background='green';

mas2[i][j].style.outline = '1px solid #000';

mas2[i][j].style.display = 'inline-block';

mas2[i][j].innerHTML = apm[count];

count++;

}

br1 = document.createElement('br');

document.body.appendChild(br1);

}

count = 0;

for(var i=0;i<srav;i++){

c\_apm[i] = [];

for(var j=0;j<val2;j++){

c\_apm[i][j] = apm[count];

count++;

}

}

for(var i=0;i<srav;i++){

count = 0;

id[i] = [];

for(var j=0;j<val2;j++){

if(c\_apm[i][j] == 1){

id[i].push(j);

count++;

}

}

ed[i] = count;

}

count = 0;

for(var i=0;i<c\_apm.length;i++){

for(var j=0;j<val4;j++){

if(i == j)continue;

count=0;

for(var k=0;k<c\_apm.length;k++){

if(c\_apm[i][k] == 1 && c\_apm[j][k] == 1){

count++;

if(ed[i] <= ed[j]){

if(count == ed[i]){

del.push(j);

for(var n=0;n<val2;n++){

c\_apm[j][n] = 9;

}

}

}

else{

if(count == ed[j])

{

del.push(i);

for(var n=0;n<val2;n++){

c\_apm[i][n] = 9;

}

}

}

}

}

}

}

//}

var flag = 0;

for(var i=0;i<c\_apm.length;i++){

for(var j=0;j<val2;j++){

if(c\_apm[i][j]==9){

c\_apm.splice(i,1);

if(i != 0)i--;

continue;

}

}

}

mas2=[];

count = 0;

for(var i=0; i<c\_apm.length;i++)

{

mas2[i] = [];

br1 = document.createElement('br');

document.body.appendChild(br1);

for(var j=0;j<val2;j++)

{

div = document.createElement('div');

document.body.appendChild(div);

mas2[i][j] = div;

mas2[i][j].style.width='30px';

mas2[i][j].style.height='30px';

mas2[i][j].style.background='green';

mas2[i][j].style.outline = '1px solid #000';

mas2[i][j].style.display = 'inline-block';

mas2[i][j].innerHTML = c\_apm[i][j];

count++;

}

}

for(var i=0;i<del.length;i++){

p = document.createElement('p');

document.body.appendChild(p);

p.innerHTML = "Удалена строка "+ (del[i]+1);

}

del =[];

copy = [];

for(var i=0;i<c\_apm.length;i++){

del[i]=[];copy[i] = []; count=0;

for(var j=0;j<val2;j++){

if(c\_apm[i][j] == 1){

copy[i][count] = [];

copy[i][count].push(j+1);

del[i].push("P"+(j+1));

count++;

}

}

}

apm = [];

for(var i=0;i<del.length;i++){

apm[i]=[];

apm[i].push("(");

for(var j=0;j<del[i].length;j++){

if(j == del[i].length-1){

apm[i].push(del[i][j]);

}

else{

apm[i].push(del[i][j]);

apm[i].push("v");

}

}

if(i == del.length - 1){

apm[i].push(")");

}

else{

apm[i].push(")");

apm[i].push("&");

}

}

var p = document.createElement('p');

document.body.appendChild(p);

p.innerHTML = apm;

var f = copy.length;

for(var i=0;i<f-1;i++){

copy.push([]);count=0;

for(var j=0;j<copy[i].length;j++){

for(var k = 0; k<copy[copy.length-2].length;k++){

copy[copy.length-1].push([]);

count1 = 0;

for(var u=0;u<copy[i][j].length;u++){

for(var y=0;y<copy[copy.length-2][k].length;y++){

copy[copy.length-1][count][count1] = copy[i][j][u];

count1++;

copy[copy.length-1][count][count1] = copy[copy.length-2][k][y];

count1++;

}

}

count++;

}

}

}

del =[];

for(var i=0;i<copy[copy.length-1].length;i++){

del[i] = [];

for(var j = 0;j<copy[copy.length-1][i].length;j++){

del[i][j] = copy[copy.length-1][i][j];

}

}

for(var i=0;i<del.length;i++){

del[i].sort();

}

for(var i =0;i<del.length;i++){

for(var j =0;j<del[i].length;j++){

for(var k =j+1;k<del[i].length;k++){

if(del[i][j] == del[i][k]){

del[i].splice(k,1);

if(i !=0) i--;

if(j!=0)j--;

}

}

}

}

c\_apm = [];

for(var i=0;i<del.length;i++){

c\_apm[i] = del[i].length;

}

for(var i=0;i<del.length;i++){

for(var j=0;j<del.length;j++){

if(c\_apm[i] < c\_apm[j]){

var l = c\_apm[i];

c\_apm[i] = c\_apm[j];

c\_apm[j] = l;

l = del[i];

del[i] = del[j];

del[j] = l;

}

}

}

c\_apm =[];flag = 0;

while(flag == 0){

flag=1;

for(var i =0;i<del.length;i++){

for(var j=i+1;j<del.length;j++){

count=0;

for(var k=0;k<del[i].length;k++){

for(var h =0;h<del[j].length;h++){

if(del[i][k] == del[j][h]){

count++;

break;

}

}

}

if(del[i].length == count){

del.splice(j,1);

if(i == 0 || j==0)flag =0;

}

}

}

}

apm=[];

for(var i=0;i<del.length;i++){

apm[i]=[];

for(var j=0;j<del[i].length;j++){

if(j == 0){

apm[i][j] = "(P"+del[i][j]+" v";

}

else if(j == del[i].length-1 && i == del.length-1){

apm[i][j] = "P"+del[i][j]+")";

}

else if(j == del[i].length-1){

apm[i][j] = "P"+del[i][j]+") &";

}

else{

apm[i][j] = "P"+del[i][j]+" v";

}

}

}

var p = document.createElement('p');

document.body.appendChild(p);

p.innerHTML = "После преобразований " + apm;

c\_apm =[];

for(var i =0;i<val2;i++){

count=0; c\_apm[i]=[];

for(var j=0;j<del.length; j++){

for(var k =0;k<del[j].length;k++){

if((i+1) == del[j][k]){

count++;

}

}

}

c\_apm[i]=count+"/"+del.length;

}

for(var i =0;i<val2;i++){

var p = document.createElement('p');

document.body.appendChild(p);

p.innerHTML = "R("+(i+1)+")= " + c\_apm[i];

}

copy=[];

count=0;

for(var i=0;i<val3.length;i++){

copy.push([]);

for(var u=0;u<val3[i];u++){

copy[i].push([]);

for(var j =0;j<del.length;j++){

copy[i][u].push([]);

for(var k=0;k<del[j].length;k++){

copy[i][u][j].push(mas3[count][del[j][k]-1]);

}

}

count++;

}

}

apm =[];

for(var i=0;i<del.length;i++){

apm.push([]);

for(var j=0;j<del[i].length;j++){

apm[i].push(d[del[i][j]-1]\*1);

}

}

c\_apm=[];

for(var i=0;i<copy.length;i++){

c\_apm[i] = [];

for(var j=0;j<apm.length;j++){

count1 =0;

for(var k=0;k<copy[i].length;k++){

count =0;

for(var u=0;u<apm[j].length;u++){

if(copy[i][k][j][u] == apm[j][u]){

count++;

}

}

if(count == apm[j].length){

count1++;

c\_apm[i][j] = count1;

}

else if(count1 == 0){

c\_apm[i][j] = 0;

}

}

}

}

for(var i=0; i<del.length+1;i++){

mas2[i] = [];

for(var j=0;j<val1+1;j++){

div = document.createElement('div');

document.body.appendChild(div);

mas2[i][j] = div;

mas2[i][j].style.width='60px';

mas2[i][j].style.height='30px';

mas2[i][j].style.background='green';

mas2[i][j].style.outline = '1px solid #000';

mas2[i][j].style.display = 'inline-block';

if(i == 0){

if(j == 0){

mas2[i][j].innerHTML = "";

}

else{

mas2[i][j].innerHTML = j;

}

}

if(j == 0){

mas2[i][j].innerHTML = del[i-1];

}

else if(i != 0 && j != 0){

mas2[i][j].innerHTML = c\_apm[j-1][i-1];

}

}

br1 = document.createElement('br');

document.body.appendChild(br1);

}

var general=0;

for(var i=0;i<c\_apm.length;i++){

for(var j=0;j<c\_apm[i].length;j++){

general+=c\_apm[i][j];

}

var p = document.createElement('p');

document.body.appendChild(p);

p.innerHTML = (i+1)+" групаа " + general;

}

}