МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий

Кафедра «Информационные системы»

**ОТЧЕТ**

о выполнении лабораторной работы №1

по дисциплине “Теория принятия решений”

Выполнил: ст. гр. ИС/б-20-2-о

Белик Г. М.

Проверил: доц. каф.

«Информационные системы»

Кротов К. В.

Севастополь

2022

**ЛАБОРАТОРНАЯ РАБОТА №1**

**«Исследование применения аппарата бинарных отношений для решения задачи выбора альтернатив»**

**Цель работы**

Исследовать применение аппарата бинарных отношений при принятии решений по выбору альтернатив.

**Задачи**

Вариант 1. Выполнить разработку программы, реализующей определение множества максимальных элементов MaxR, руководствуясь заданной формой графа отношений. При разработке программы использовать приведенные в теоретическом введении правила формирования множества MaxR. При разработке программы использовать следующий вид графа отношений между решениями множества Х.

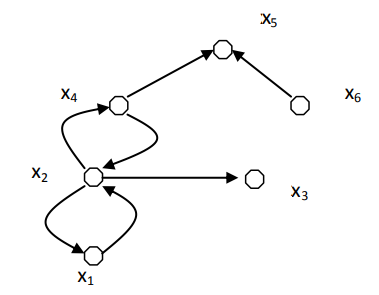
****

Рисунок 1 – Граф отношений

1. Для задания на работу, связанного с формированием подмножества максимальных

элементов MaxR множества Х, необходимо по заданному варианту графа отношений предпочтения между решениями сформировать матрицу А отношения R (где R– отношение φ ). При этом убедиться, что первый элемент множества Х является строго независящим от других решений.

1. Выполнить формирование множества MaxR вручную для заданного вида графа и

соответствующего ему вида матрицы А.

1. Выполнить формирование программного кода соответствующей процедуры определения.

множества MaxR, при этом возможно руководствоваться ориентировочным видом процедуры определения этого множества, предложенным в теоретическом введении данной лабораторной работы.

1. Выполнить вывод результатов работы процедуры и сравнить полученные в процедуре.

результаты с результатами, сформированными аналитически.

1. Изменить исходные данные программы, используя графы отношений из примера 5.

Проверить получаемые с использованием процедуры результаты с аналитическим результатами, формируемыми для этих графов.

**Ход работы**

Для начала по заданному варианту графа отношений предпочтения между решениями была сформирована матрица А отношения R.

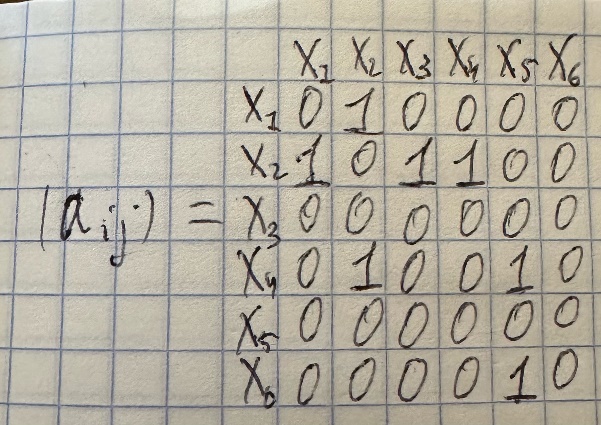
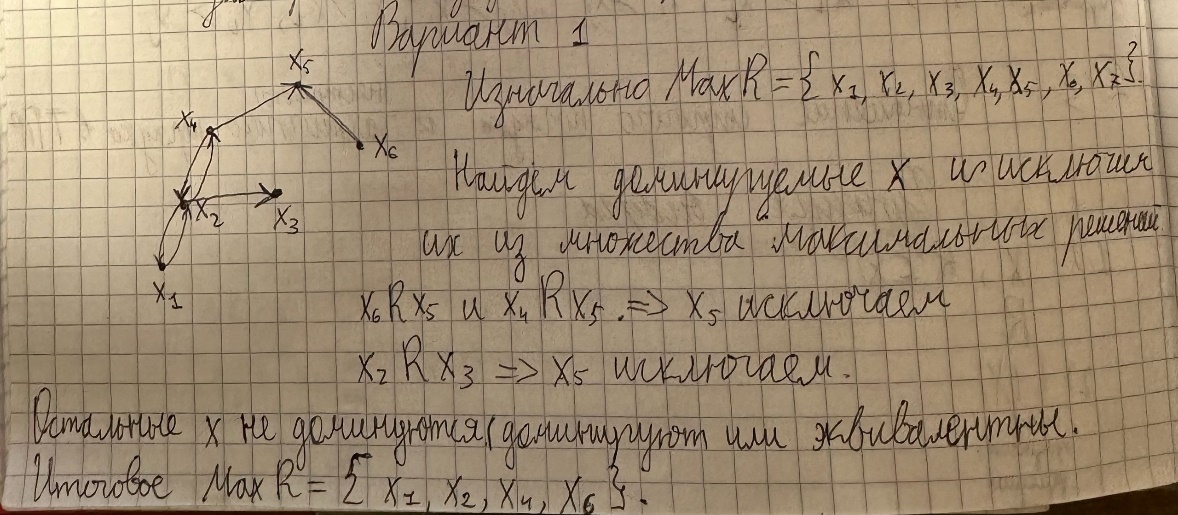


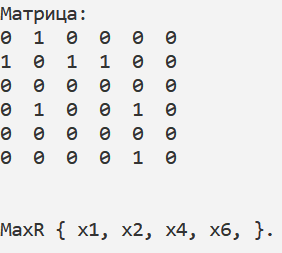
Рисунок 2 – Матрица отношений

Затем вручную было выполнено формирование множества MaxR для заданного вида графа и соответствующего ему вида матрицы А.

 Рисунок 3 – Расчеты множества MaxR

Следующим шагом был сформирован программный код на языке C++ соответствующей процедуры определения множества MaxR, представленный в приложении А.

После этого выведем результат работы процедуры.

 Рисунок 4 – Матрица отношений и найденное ее максимальное множество

Результаты, полученные в процедуре аналогичны сформированным аналитически.

Изменим исходные данные программы и увидим, что для них программа также находит верные значения максимального множества.

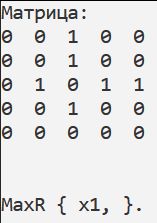


Рисунок 5 – Матрица отношений и найденное ее максимальное множество

**Выводы**

В ходе выполнения лабораторной работы был исследован аппарат бинарных отношений при принятии решений по выбору альтернатив. По заданному графу отношений предпочтения была сформирована матрица отношений R, аналитически нашли множество максимальных элементов MaxR, был написан программный код для нахождения MaxR, сопоставлены полученные значения, и произвели повторную проверку правильности работы программы на другом графе бинарных отношений.

**ПРИЛОЖЕНИЕ А**

#include <iostream>

#include <locale>

using namespace std;

int main()

{

setlocale(LC\_ALL, "");//Заполнение матрицы

int n; cout << " Введите размерность матрицы:\n"; cin >> n;

int\*\* a = new int\* [n];

cout << " Заполните масив из " << n \* n << " элементов:\n";

for (int i = 0; i < n; i++)

{

a[i] = new int[n];

for (int j = 0; j < n; j++)

{

cout << "massiv[" << i + 1 << "][" << j + 1 << "] = ";

cin >> a[i][j];

}

}

// Печать матрицы

cout << endl << "Матрица:" << endl;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

cout << a[i][j] << " ";

cout << endl;

}

cout << endl;

// Расчеты

int\* MaxR = new int[n];

for (int i = 0; i < n; i++)

MaxR[i] = 1;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

{

if (a[i][j] == 1)

{

if (a[j][i] == 0)

MaxR[j] = 0;

}

}

}

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

{

if (a[i][j] == 1 && MaxR[i] == 0)

MaxR[j] = 0;

}

}

cout << endl;

cout << "MaxR { ";

// Вывод ответа

for (int i = 0; i < n; i++)

{

if (MaxR[i] == 1)

cout << "x" << i + 1 << ", ";

}

cout << "}." << endl;

}