Министерство образования и науки Российской Федерации

Севастопольский национальный технический университет

Кафедра ИС

ОТЧЕТ

по лабораторной работе № 1

«**Исследование применения аппарата бинарных**

**отношений для решения задачи выбора альтернатив**»

Вариант 2

|  |  |
| --- | --- |
|  |  |

Выполнил:

ст. гр. И-33д

**Иванов А.В.**

Проверил:

доцент кафедры ИС

**Кротов К.В.**

Севастополь

2014

1. **ЦЕЛЬ РАБОТЫ**

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

1. **ПОСТАНОВКА ЗАДАЧИ**

**Вариант 2.** Выполнить разработку программы, реализующей определение упорядоченного множества решений для множества *Х*, руководствуясь заданной формой графа отношений. При разработке программы использовать приведенные в теоретическом введении правила формирования множества с учетом рассмотрения вершин-источников на каждом шаге алгоритма. При формировании упорядоченного множества решений указывать номер яруса, на котором находятся решения. Определить эффективные решения. При разработке программы использовать следующий вид графа отношений между решениями множества *Х*.

x4

x3

x5

x1

x2

x6

x7

Рисунок 1 — граф 1

x4

x3

x5

x1

x2

Рисунок 2 — граф 2

1. **ХОД РАБОТЫ**
   1. Аналитическое решение

Приведем матрицу инцидентности графа 1:

Упорядоченное множество решений будет иметь вид:

Таблица 1 — упорядоченное множество решений для графа 1

|  |  |
| --- | --- |
| Ярус | Эффективное решение |
| 1 | x1, x5 |
| 2 | x4 |
| 3 | x2 |
| 4 | x3 |
| 5 | x6 |
| 6 | x7 |

Приведем матрицу инцидентности графа 2:

Упорядоченное множество решений для графа 2 будет иметь вид:

|  |  |
| --- | --- |
| Ярус | Эффективное решение |
| 1 | x1 |
| 2 | x2 |
| 3 | x3 |
| 4 | x4, x5 |

* 1. Программная реализация
     1. Текст программы

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

int n;

ifstream fin("in.txt");

fin>>n;

int matr[n][n];

int MaxR[n];

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

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

fin >> matr[i][j];

int count = 0;

int sum;

int floor = 1;

int t\_c = 0;

while (count < n) {

t\_c = count;

cout << endl << "Ярус " << floor++ << ": ";

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

sum = 0;

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

sum += matr[i][j];

if (sum == 0) {

cout << "x" << j + 1<<" ";

MaxR[count] = j;

count++;

}

}

if (t\_c == count) { // если не найдена вершина источник

cout << "Невозможно определить вершину-источник"<<endl;

break;

} else {

if ( count - t\_c > 1)

cout << " Решения несравнимы ";

}

for (int q = count - 1; q >= 0; q--)

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

matr[MaxR[q]][j] = 0;

for (int q = count - 1; q >= 0; q--)

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

matr[i][MaxR[q]] = 1;

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

cout<<endl;

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

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

}

}

cout << endl;

return 0;

}

* + 1. Тестовые примеры

Программа выводит информацию об эффективном решении и матрицу инцидентности.

Протестируем работу программы, введя матрицы инцидентности для первого и второго графов.

Результат работы программы для графа 1:

*Ярус 1: x1 x5 Решения несравнимы*

*1 0 0 0 1 0 0*

*1 0 1 0 1 0 0*

*1 0 0 0 1 1 1*

*1 1 0 0 1 1 0*

*1 0 0 0 1 0 0*

*1 0 0 0 1 0 1*

*1 0 0 0 1 0 0*

*Ярус 2: x4*

*1 0 0 1 1 0 0*

*1 0 1 1 1 0 0*

*1 0 0 1 1 1 1*

*1 0 0 1 1 0 0*

*1 0 0 1 1 0 0*

*1 0 0 1 1 0 1*

*1 0 0 1 1 0 0*

*Ярус 3: x2*

*1 1 0 1 1 0 0*

*1 1 0 1 1 0 0*

*1 1 0 1 1 1 1*

*1 1 0 1 1 0 0*

*1 1 0 1 1 0 0*

*1 1 0 1 1 0 1*

*1 1 0 1 1 0 0*

*Ярус 4: x3*

*1 1 1 1 1 0 0*

*1 1 1 1 1 0 0*

*1 1 1 1 1 0 0*

*1 1 1 1 1 0 0*

*1 1 1 1 1 0 0*

*1 1 1 1 1 0 1*

*1 1 1 1 1 0 0*

*Ярус 5: x6*

*1 1 1 1 1 1 0*

*1 1 1 1 1 1 0*

*1 1 1 1 1 1 0*

*1 1 1 1 1 1 0*

*1 1 1 1 1 1 0*

*1 1 1 1 1 1 0*

*1 1 1 1 1 1 0*

*Ярус 6: x7*

*1 1 1 1 1 1 1*

*1 1 1 1 1 1 1*

*1 1 1 1 1 1 1*

*1 1 1 1 1 1 1*

*1 1 1 1 1 1 1*

*1 1 1 1 1 1 1*

*1 1 1 1 1 1 1*

Результат работы программы для графа 2:

*Ярус 1: x1*

*1 0 0 0 0*

*1 0 1 1 0*

*1 0 0 1 1*

*1 0 0 0 0*

*1 0 0 0 0*

*Ярус 2: x2*

*1 1 0 0 0*

*1 1 0 0 0*

*1 1 0 1 1*

*1 1 0 0 0*

*1 1 0 0 0*

*Ярус 3: x3*

*1 1 1 0 0*

*1 1 1 0 0*

*1 1 1 0 0*

*1 1 1 0 0*

*1 1 1 0 0*

*Ярус 4: x4 x5 Решения несравнимы*

*1 1 1 1 1*

*1 1 1 1 1*

*1 1 1 1 1*

*1 1 1 1 1*

*1 1 1 1 1*

Если вершину-источник определить невозможно, программа выдает соответствующее сообщение и прекращает работу.

Введем матрицу инцидентности графа:

Результат работы программы:

*Ярус 1: Невозможно определить вершину-источник*

**Выводы:**

В ходе работы написана и протестирована программа реализующей определение упорядоченного множества решений для множества *Х,* а так же приведено аналитическое решение. Программа показала свою работоспособность, которая была подтверждена вышеуказанными тестовыми примерами. Результаты программного и аналитического решения полностью совпали.