**Московский Авиационный Институт**

**(Национальный Исследовательский Университет)**

**Факультет информационных технологий и прикладной математики**

**Кафедра математической кибернетики**

**Курсовая работа**

**по курсу**

**«Дискретная математика»**

**2 семестр**

**Тема:**

**«Нахождение наименьшего покрытия простого графа»**

Студент: Дубинин А. О.

Группа: М8О-103Б

Руководитель: доцент Яшина Н.П.

Оценка

Дата

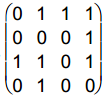
**Москва**

**2018**

**Задание.**

**Вариант 6.**

1. Определить для орграфа, заданного матрицей смежности:



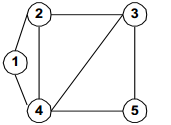
а) матрицу односторонней связности;

б) матрицу сильной связности;

в) компоненты сильной связности;

г) матрицу контуров.

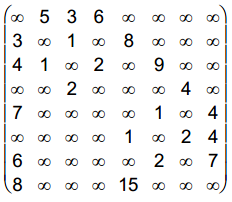
1. Используя алгоритм Терри, определить замкнутый маршрут, проходящий ровно по два раза (по одному в каждом направлении) через каждое ребро графа.

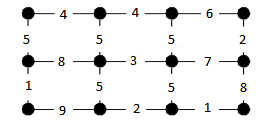


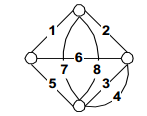
1. Используя алгоритм “фронта волны”, найти все минимальные пути из первой вершины в последнюю орграфа, заданного матрицей смежности.



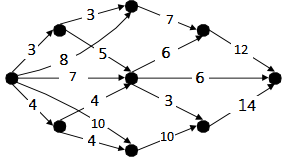
1. Используя алгоритм Форда, найти минимальные пути из первой вершины во все достижимые вершины в нагруженном графе, заданном матрицей длин дуг.



1. Найти остовное дерево с минимальной суммой длин входящих в него ребер.
2. Пусть каждому ребру неориентированного графа соответствует некоторый элемент электрической цепи. Составить линейно независимые системы уравнений Кирхгофа для токов и напряжений. Пусть первому и пятому ребру соответствуют источники тока с ЭДС 𝐸1 и 𝐸2, а остальные элементы являются сопротивлениями. Используя закон Ома, и, предполагая внутренние сопротивления источников тока равными нулю, получить систему уравнений для токов.



1. Построить максимальный поток по транспортной сети.



1. Нахождение наименьшего покрытия простого графа.

**Задание №8**

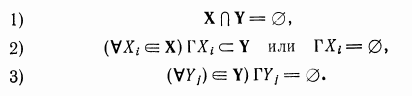
**Нахождение наименьшего покрытия простого графа.**

**Теоретические сведения. Описание алгоритма.**

Рассмотрим простой граф G = (V, E).

**Определение 1** — Простым графом называют такой граф

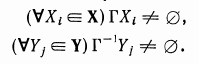
G = (X Y, Г), что



Простой граф будем обозначать G = (X Y Г).

**Определение 2** — Покрытием простого графа G = (X Y Г) называют такое подмножество дуг W U, что любая вершина графа инцидентна по крайней мере одной дуге из W.

Для того чтобы простой граф обладал покрытием, необходимо и достаточно выполнение условий:



Например, множество



— покрытие простого графа на рис 1(на рис. 2 и 3 оно выделено)

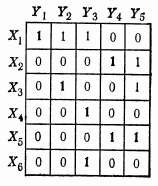
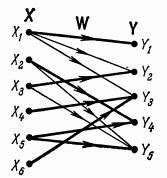
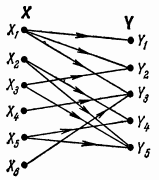


Рис. 1. Рис. 2. Рис. 3.

Исходя из булева матричного представления простого графа, можно определить покрытие как такой набор единиц, что каждая строка и каждый столбец матрицы содержат по крайней мере по одному элементу из этого набора.

**Определение 3** — Минимальное покрытие простого графа. Требуется найти с минимальным , т.е. с наименьшим числом дуг.

**Описание алгоритма, использующего булево матричное представление.**

Опишем алгоритм (рассматривая одновременно пример) позволяющий получить минимальное покрытие простого графа G = (X, Y, Г), представленного в виде булевой матрицы **||**M**||** размера (в нашем примере в качестве **||**M**||** берется матрица на рис 4)

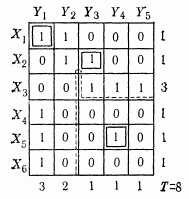
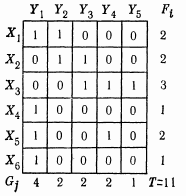


Рис. 4. Рис. 5.

1. Сопоставим каждой строке матрицы **||**M**||** число равное сумме

ее элементов, и каждому столбцу Y — число равное сумме его элементов (на рис 4 указаны справа, а внизу). Очевидно, что



1. Если



То множество дуг, соответствующих единицам, дает минимальное покрытие. Если



То заменяем последовательно в произвольном порядке нулем

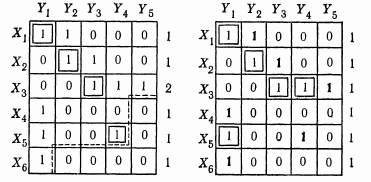


Рис. 6. Рис. 7.

каждую из единиц (условливаясь при этом писать  вместо 0), для которой

Например, заключая в квадратик 1 на местах (), (), () в матрице на рис. 4, приходим к матрице на рис. 5. Легко видеть, что эту операцию дальше продолжить нельзя.

1. В каждой строке с ищем такую неотмеченную 1, что

содержащем ее столбце найдется такая отмеченная 1, что в строке содержащей эту 1, есть либо неотмеченная 1 с либо отмеченная 1. Если то появляется возможность увеличить число отмеченных единиц. Если -й столбец содержит отмеченную 1, то в её строке ищем неотмеченную 1, в столбце которой есть отмеченная 1, и т. д.

Если, действуя по этому алгоритму, мы уже не можем больше увеличить число отмеченных единиц, то необведённые единицы дают минимальное покрытие

В нашем примере мы переходим от рис. 5 к рис. 6, а от него к рис. 7 (пунктир показывает порядок действия по 3).

**Логическая блок-схема алгоритма.**

Основные этапы работы алгоритма представлены на рисунке логической блок схемы.

Ввод данных о графе в графическом виде

Построение булевой матрицы представления простого графа c

**ИНАЧЕ** **ЕСЛИ**

,

**ЕСЛИ**

,

Заменяем в произвольном порядке нулем каждую из единиц для которой

**ИНАЧЕ**

**ПОКА**

,

В каждой строке ищем неотмеченную 1, что в содержащем ее столбце найдется такая отмеченная 1, что в строке, содержащей эту единицу, есть неотмеченная 1 с , то тогда увеличиваем число отмеченных единиц

**ЕСЛИ**

,

то множество дуг соответствующим единицам даёт мин. покрытие

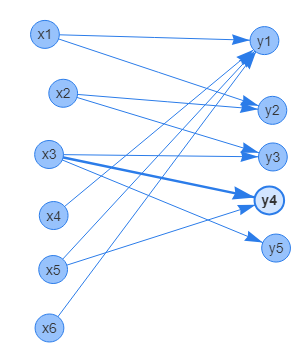
Программа написана на языке JavaScript с использованием библиотеки Vis.js.

**Оценка сложности алгоритма.**

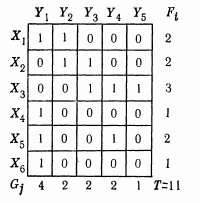
В худшем случае алгоритм работает за O(n3), где n — количество ребер в графе.

**Тестовые примеры. Скриншоты программы.**

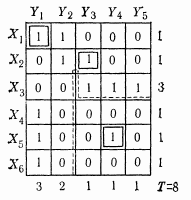
**Пример 1.** Дан простой граф:



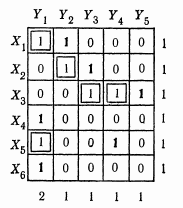
Посчитаем матрицу смежности



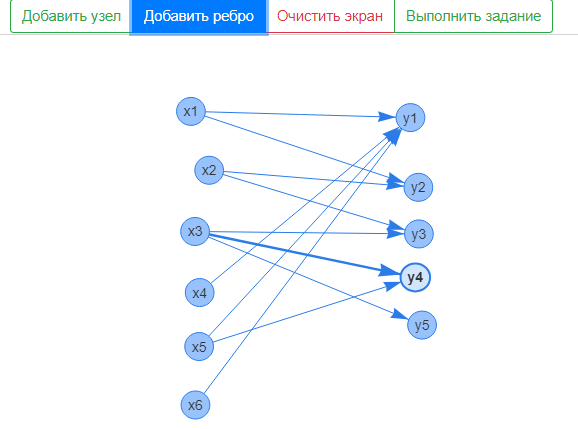
Заменяем в произвольном порядке нулем каждую из единиц для которой

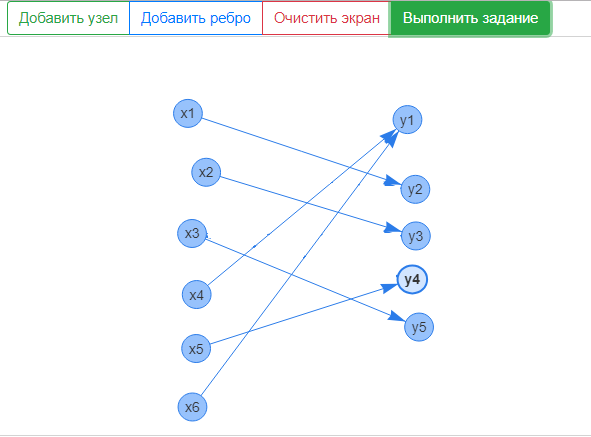


В каждой строке ищем неотмеченную 1, что в содержащем ее столбце найдется такая отмеченная 1, что в строке, содержащей эту единицу, есть неотмеченная 1 с , то тогда увеличиваем число отмеченных единиц

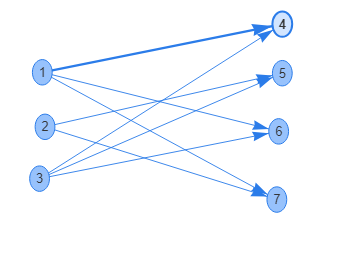


Скриншот программы:





**Пример 2.** Дан простой граф:

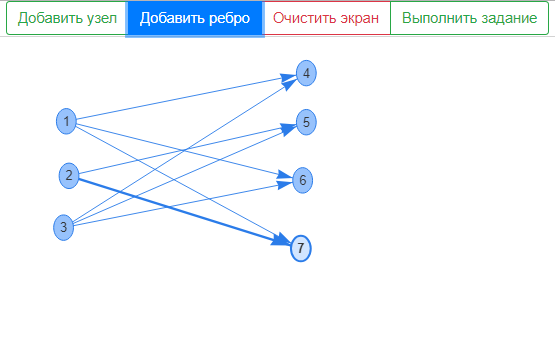


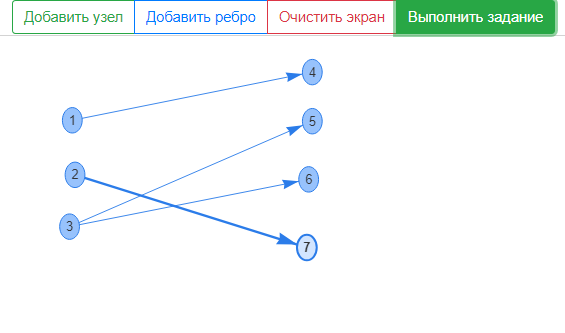
Посчитаем матрицу смежности

Заменяем в произвольном порядке нулем каждую из единиц для которой

В каждой строке ищем неотмеченную 1, что в содержащем ее столбце найдется такая отмеченная 1, что в строке, содержащей эту единицу, есть неотмеченная 1 с , то тогда увеличиваем число отмеченных единиц

Скриншот программы:





**Прикладная задача.**

Имеется n домов и k источников тепла из котельной. Изначально дома уже соединены трубами с источниками в хаотичном порядке, так, что один дом может быть соединен с k источниками (гарантируется, что каждый источник соединен хотя бы с одним домом, и каждый дом хотя бы с одним источником). Нужно демонтировать трубы, чтобы каждый источник использовался (иначе он сломается) и каждый дом обогревался (дом считается обогретым, если соединен хотя бы с одним источником). Перед вами стоит задача демонтировать максимальное кол-во труб, чтобы затраты на обогрев стали меньше.