

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Информационных технологий и управления
Кафедра Интеллектуальных информационных технологий

РАСЧЕТНАЯ РАБОТА

по дисциплине «Традиционные и интеллектуальные информационные
технологии»

на тему

**Задача нахождения всех подграфов исходного графа,
изоморфных графу-образцу**

Выполнил:

М. А. Переверзев

Студент группы
321702

Проверил:

Н. В. Малиновская

Минск 2024

1 ВВЕДЕНИЕ

Цель: Получить навыки формализации и обработки информации с использованием семантических сетей

Задача: Поиск подграфов в неориентированном графе, изоморфных графу-образцу

2 СПИСОК ПОНЯТИЙ

1. **Граф** - это совокупность непустого множества вершин и множества пар вершин

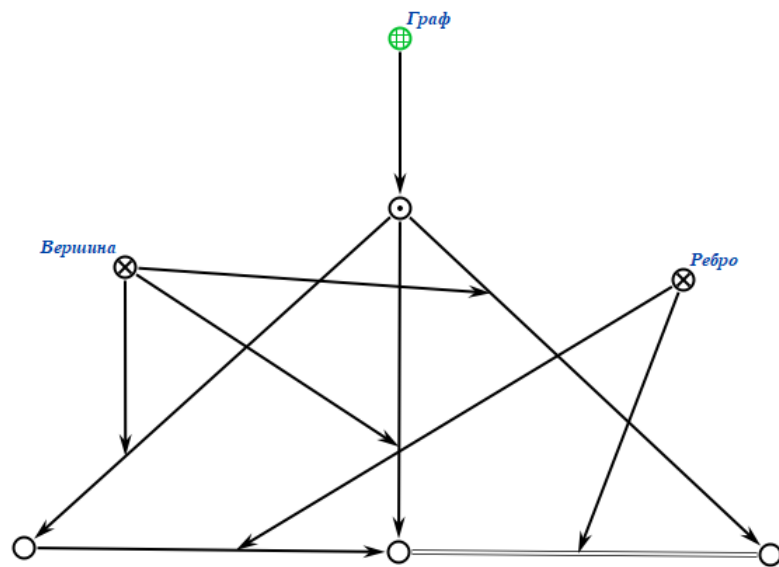


Рисунок 2.1 – Граф (полный вариант представления)

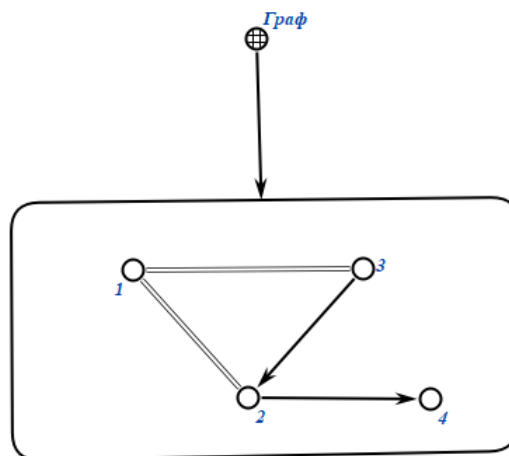


Рисунок 2.2 – Граф (сокращённый вариант представления)

2. **Неориентированный граф** - это граф, рёбра которого не имеют направления и могут быть проходимыми в обоих направлениях
- а. Вершина (относительное понятие, ролевое отношение);
 - б. Связка (относительное понятие, ролевое отношение).

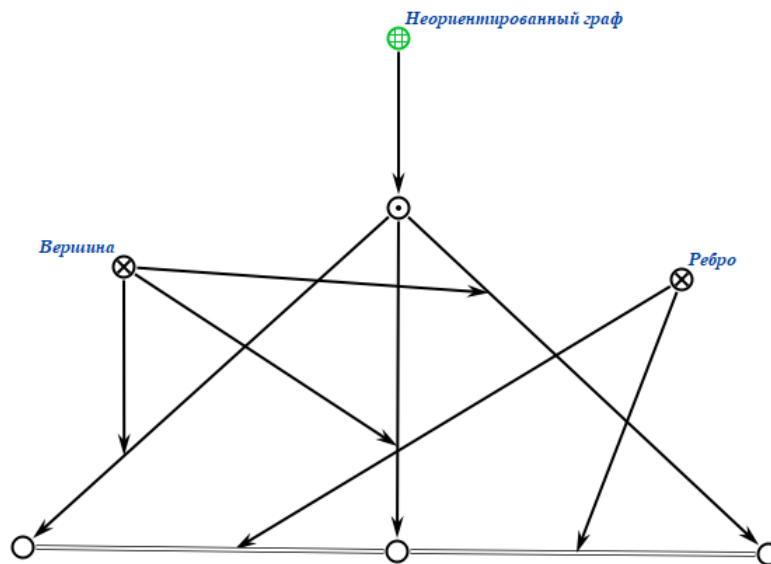


Рисунок 2.3 – Неориентированный граф (полный вариант представления)

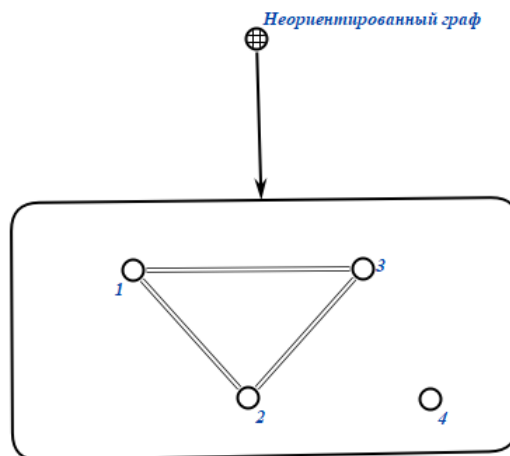


Рисунок 2.4 – Неориентированный граф (сокращённый вариант представления)

3. **Изоморфные графы** - два графа, у которых одинаковое число вершин и вершины каждого можно занумеровать так числами от 1 до n, что в первом графе две вершины соединены ребром тогда и только тогда, когда вершины с такими же номерами соединены во втором графе

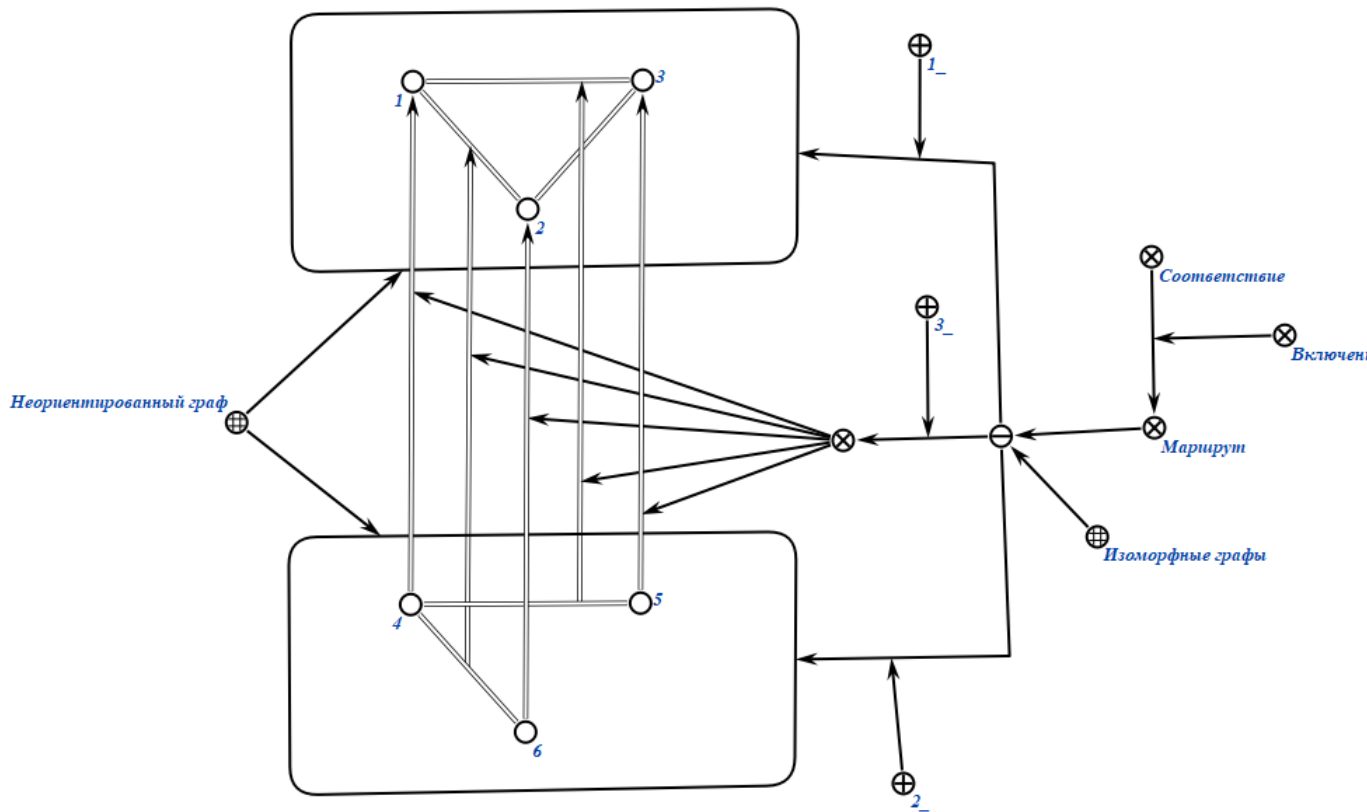


Рисунок 2.5 – Изоморфные графы

3 ТЕСТОВЫЕ ПРИМЕРЫ

Во всех тестах графы будут приведены в сокращенной форме. Далее будет подробно рассмотрен алгоритм нахождения изоморфных графов

3.1 Тест 1

Вход:

Необходимо найти все подграфы в неориентированном графе, изоморфных графу-образцу

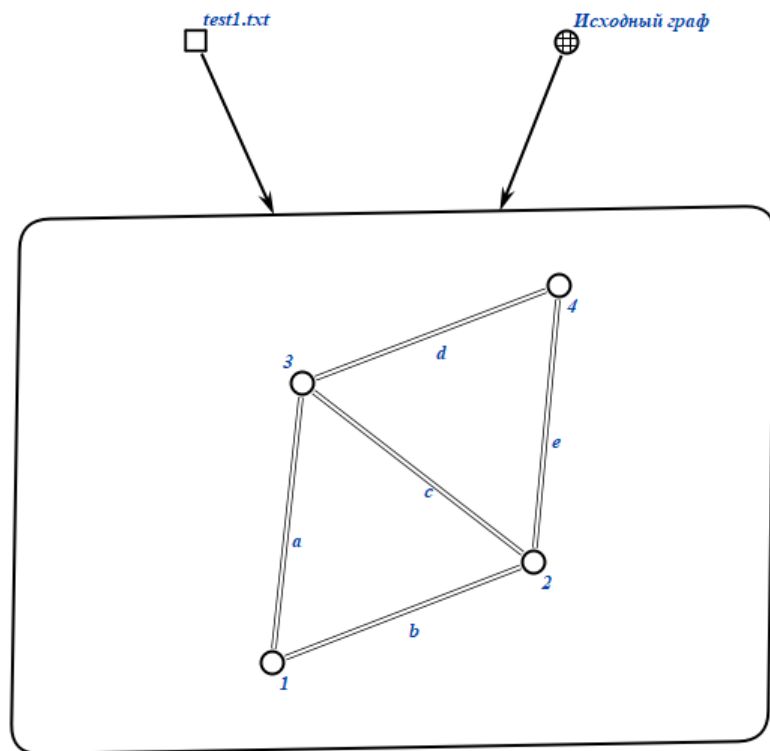


Рисунок 3.1 – Вход теста 1 (Исходный граф, в котором надо найти изоморфные подграфы)

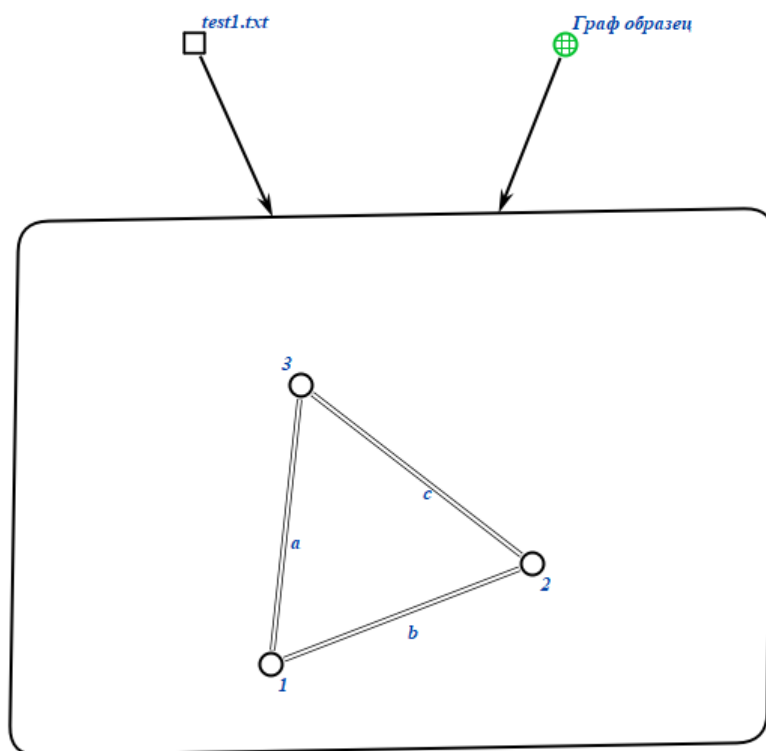


Рисунок 3.2 – Вход теста 1 (Граф образец, для которого надо найти изоморфные графы)

Выход: Результатом будет 2 графа

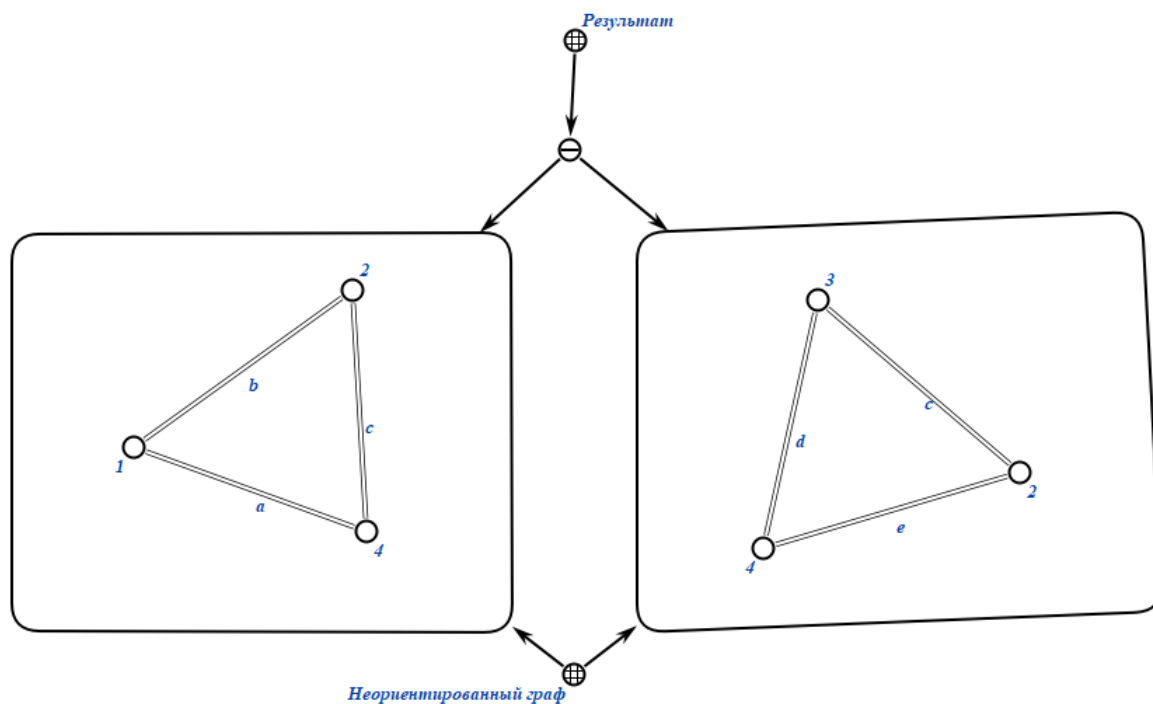


Рисунок 3.3 – Выход теста 1

3.2 Тест 2

Вход:

Необходимо найти все подграфы в неориентированном графе, изоморфных графу-образцу

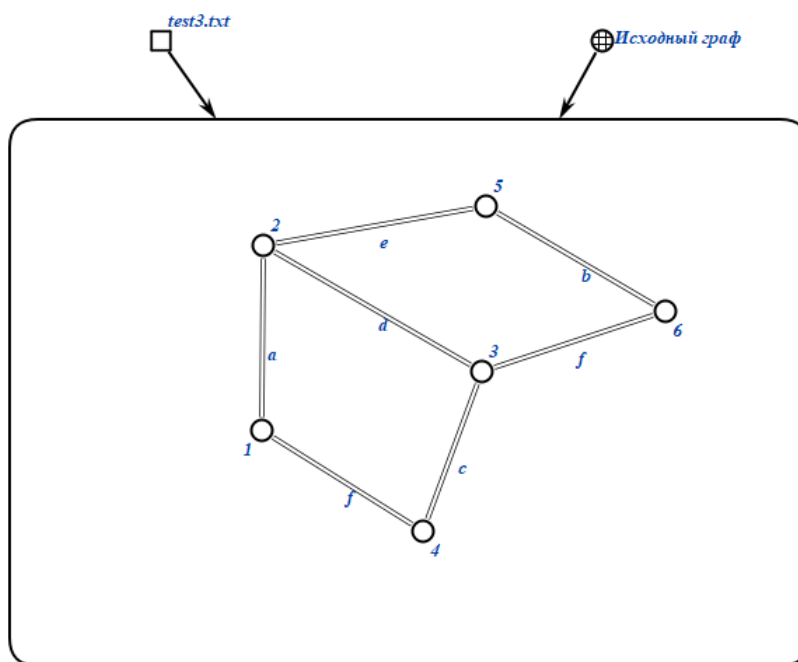


Рисунок 3.4 – Вход теста 2 (Исходный граф, в котором надо найти изоморфные подграфы)

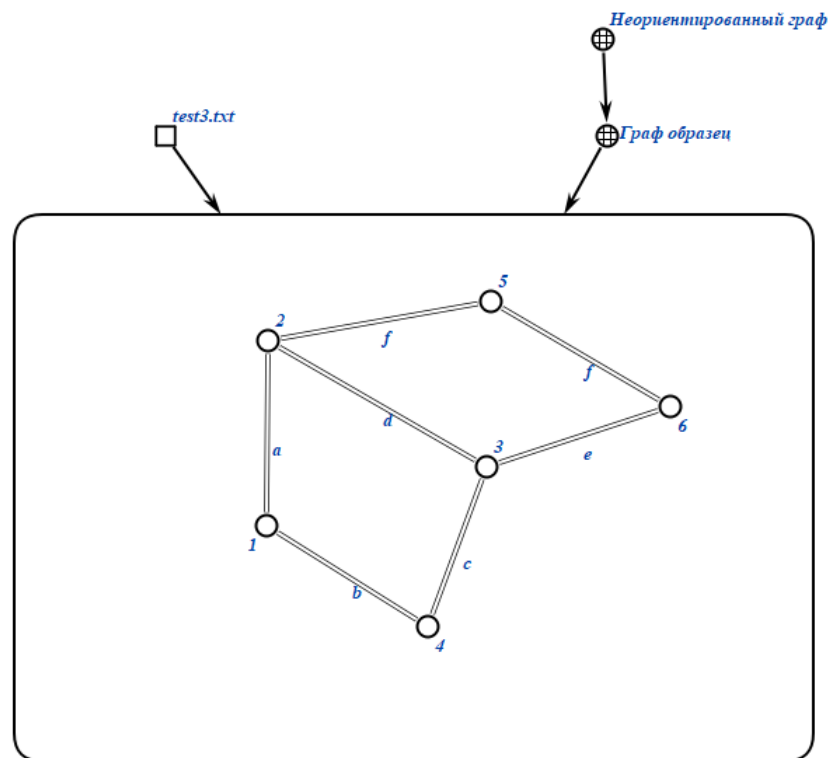


Рисунок 3.5 – Вход теста 2 (Граф образец, для которого надо найти изоморфные графы)

Выход:

Результатом будет тот же самый граф

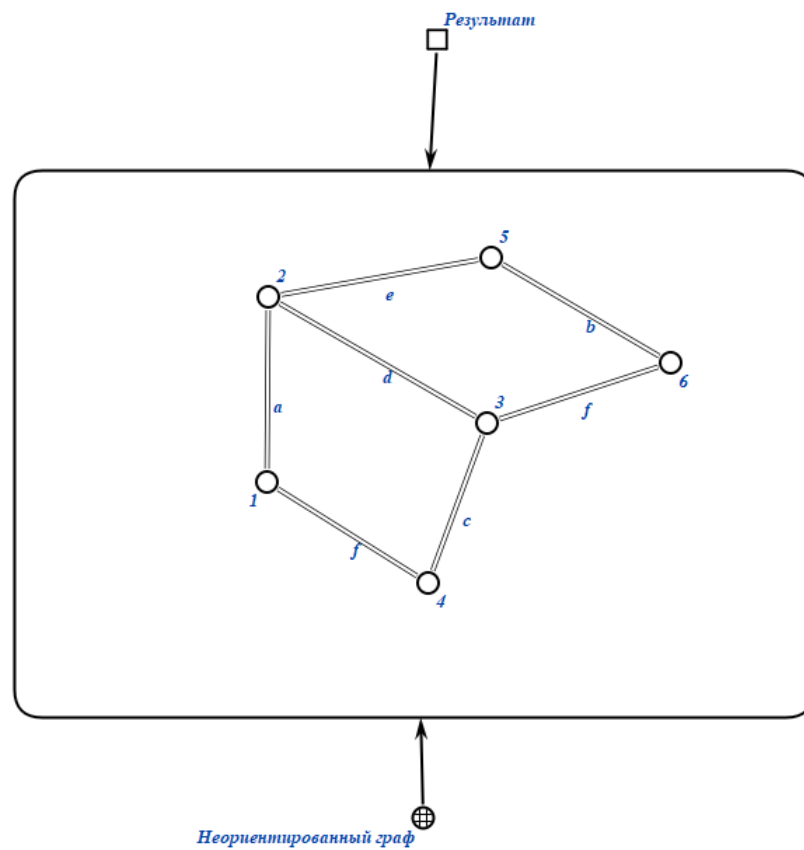


Рисунок 3.6 – Выход теста 2

3.3 Тест 3

Вход:

Необходимо найти все подграфы в неориентированном графе, изоморфных графу-образцу

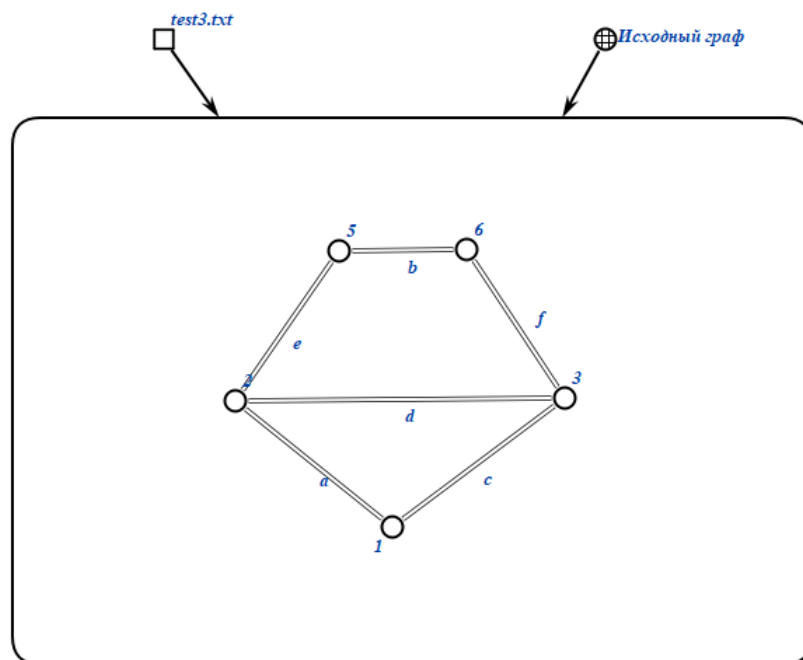


Рисунок 3.7 – Вход теста 3 (Исходный граф, в котором надо найти изоморфные подграфы)

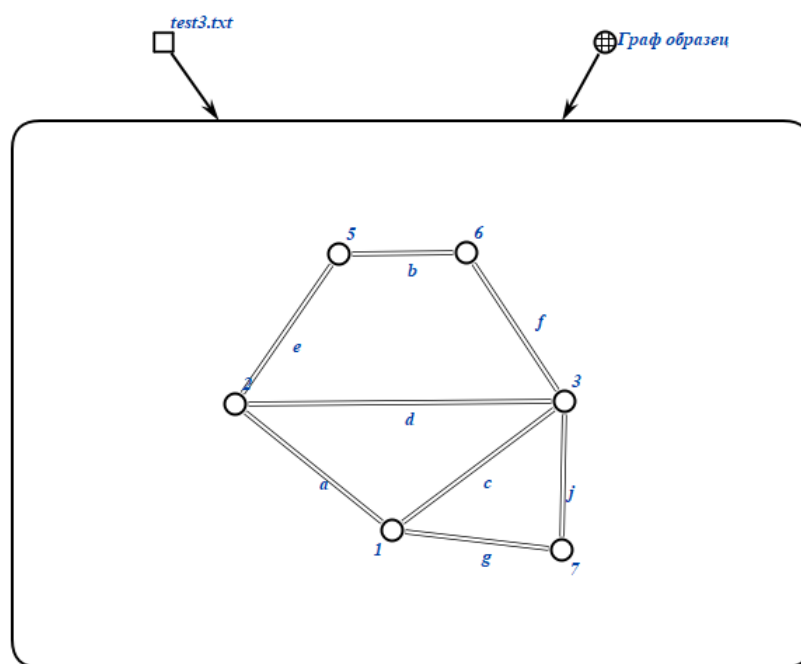


Рисунок 3.8 – Вход теста 3 (Граф образец, для которого надо найти изоморфные графы)

Выход:

Результатом пустое множество, потому что изоморфных подграфов в исходном графе не имеется

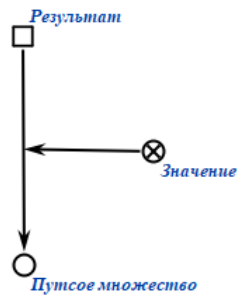


Рисунок 3.9 – Выход теста 3

3.4 Тест 4

Вход:

Необходимо найти все подграфы в неориентированном графе, изоморфных графу-образцу

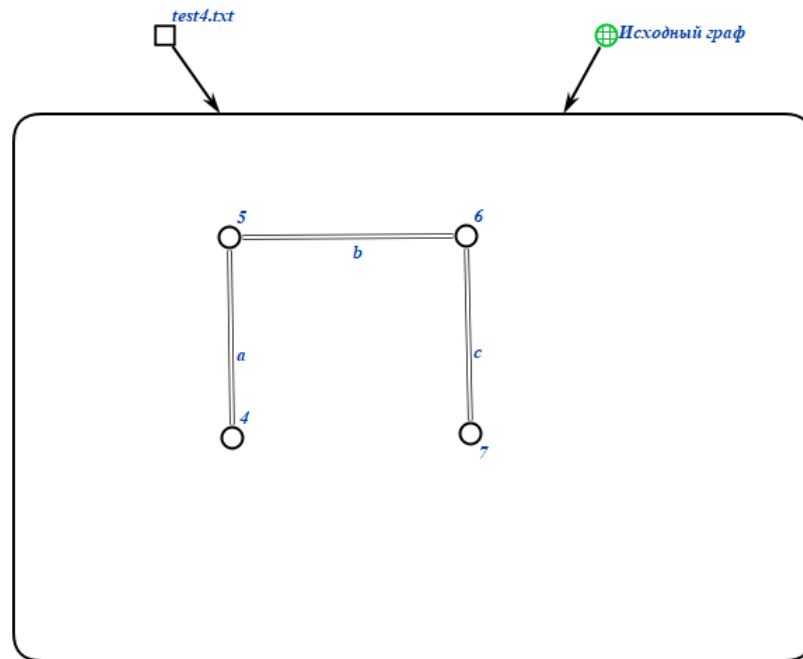


Рисунок 3.10 – Вход теста 4 (Исходный граф, в котором надо найти изоморфные подграфы)

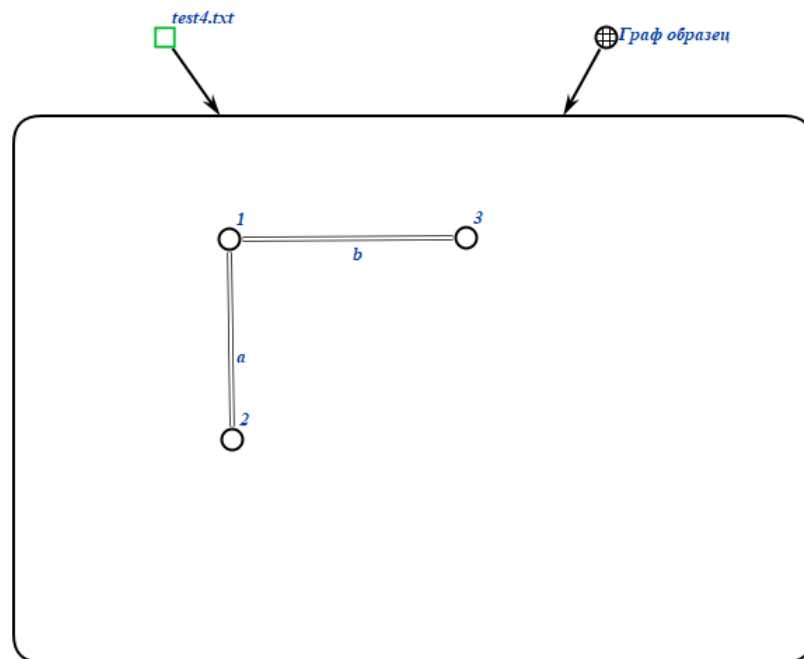


Рисунок 3.11 – Вход теста 4 (Граф образец, для которого надо найти изоморфные графы)

Выход:

Результатом будет 2 графа

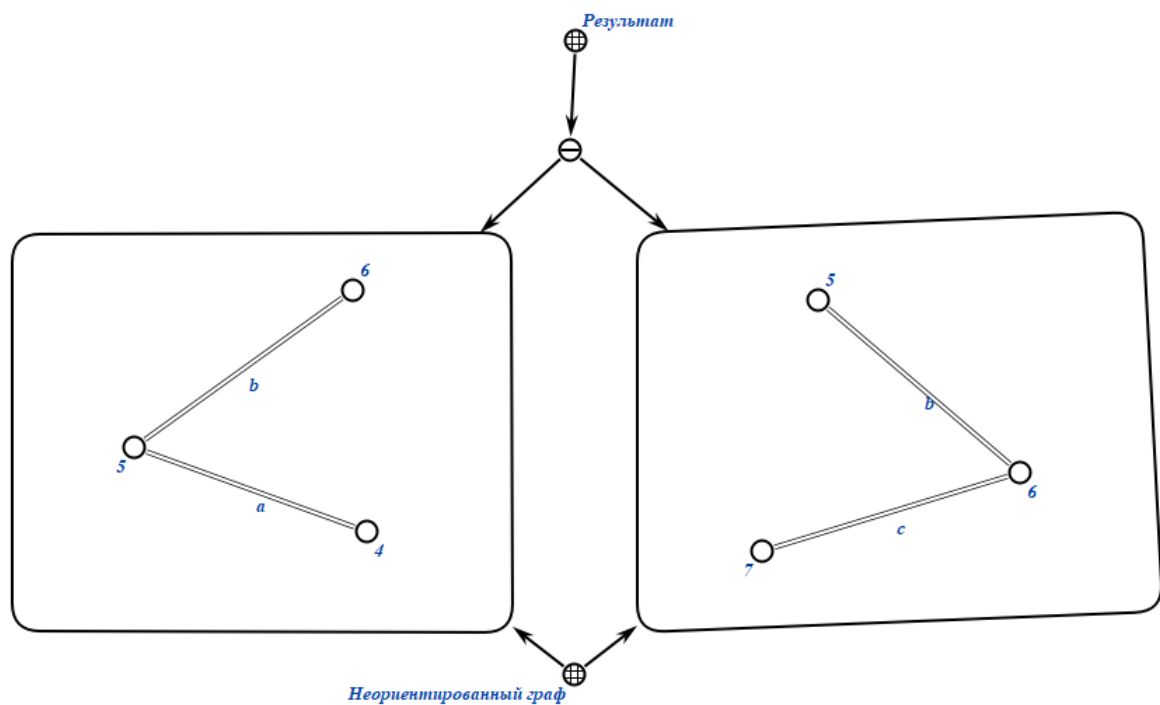


Рисунок 3.12 – Выход теста 4

4 ПРИМЕР РАБОТЫ АЛГОРИТМА В СЕМАНТИЧЕСКОЙ ПАМЯТИ

1. Задание входного графа

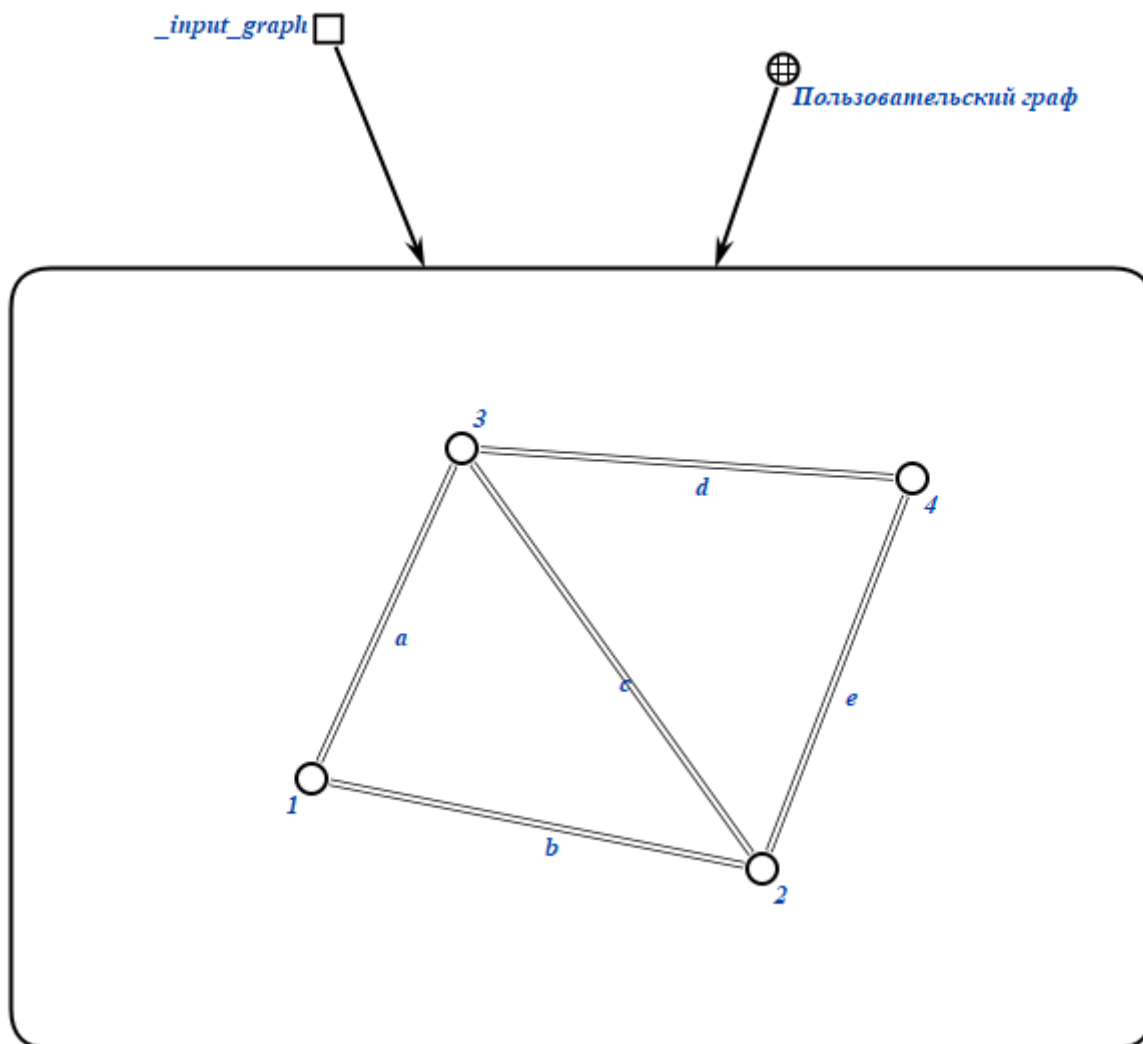


Рисунок 4.1 – Входной граф

`_input_graph` получит в качестве значения sc-узел неориентированного графа.

2. Создаём структуру `_edge_info`, в которой будут храниться данные о ребре - пара: первый элемент - название ребра, второй элемент - пара `_edge_info_nodes`, в которой будут храниться вершины, которые соединяет это ребро (`_first_node`, `_second_node`).

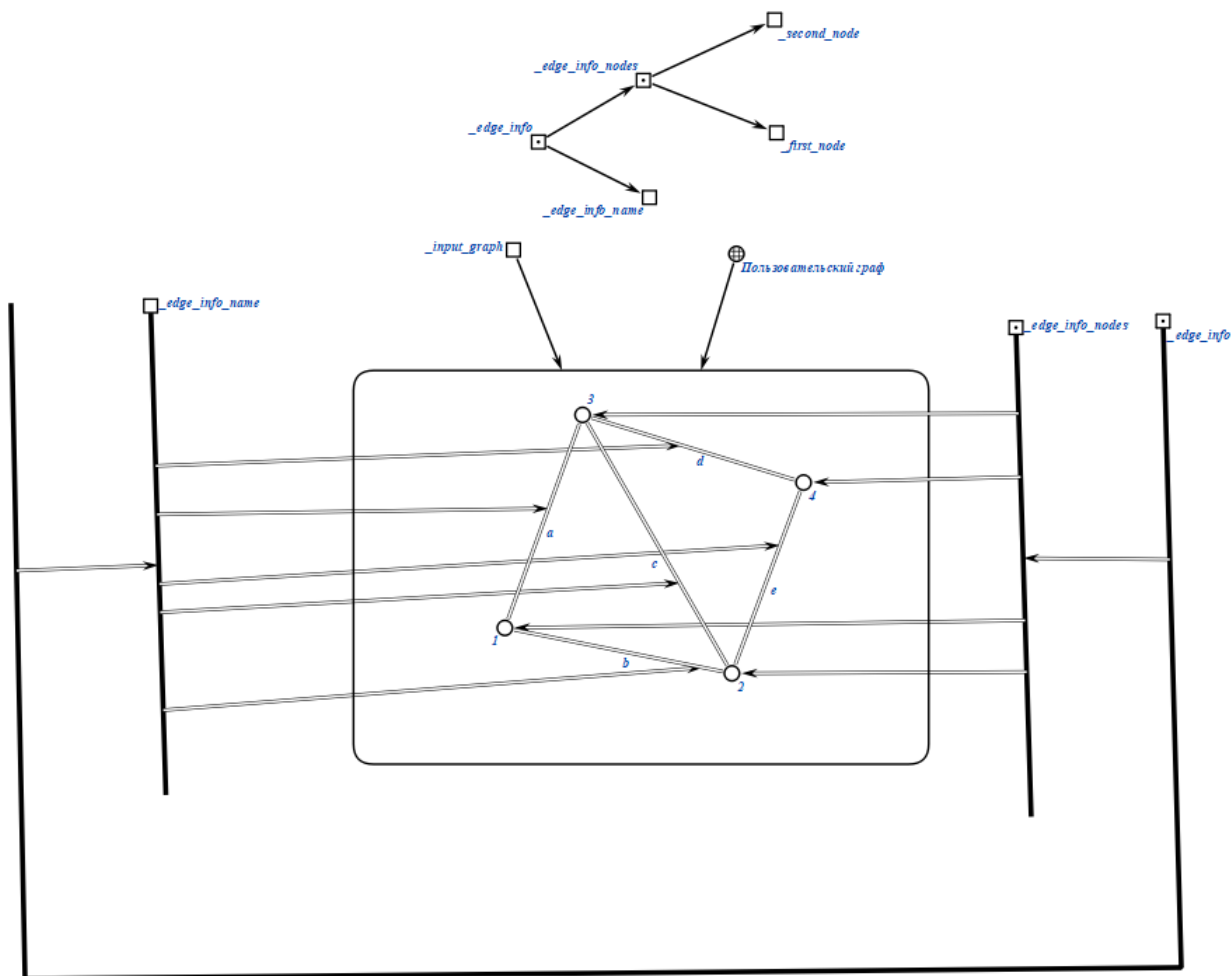


Рисунок 4.2 – Считывание данных из графа в структуру

3. Данные передаются через 2 дочерние шины: первая пропускает через себя имя ребра, вторая пропускает через себя две вершины, которые соединяются этим ребром. Затем в основную шину поступают `_edge_info_name` и `_edge_info_nodes`. Из множества таких структур состоит входной граф. Поэтому необходимо также считать количество таких структур, т.к. один из критериев изоморфности графа является одинаковое количество таких структур. Изначально счётчик равен нулю, по мере прохождения всех рёбер графа, счётчик будет изменяться

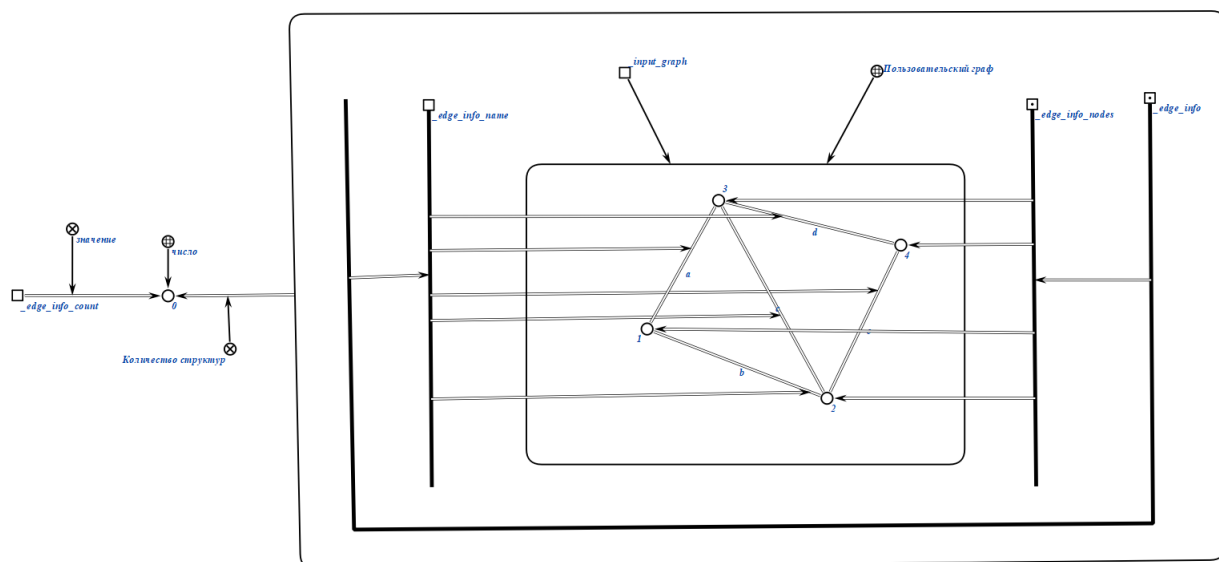


Рисунок 4.3 – Считывание количество структур в переменную

Для графа-образца алгоритм получения данных идентичен

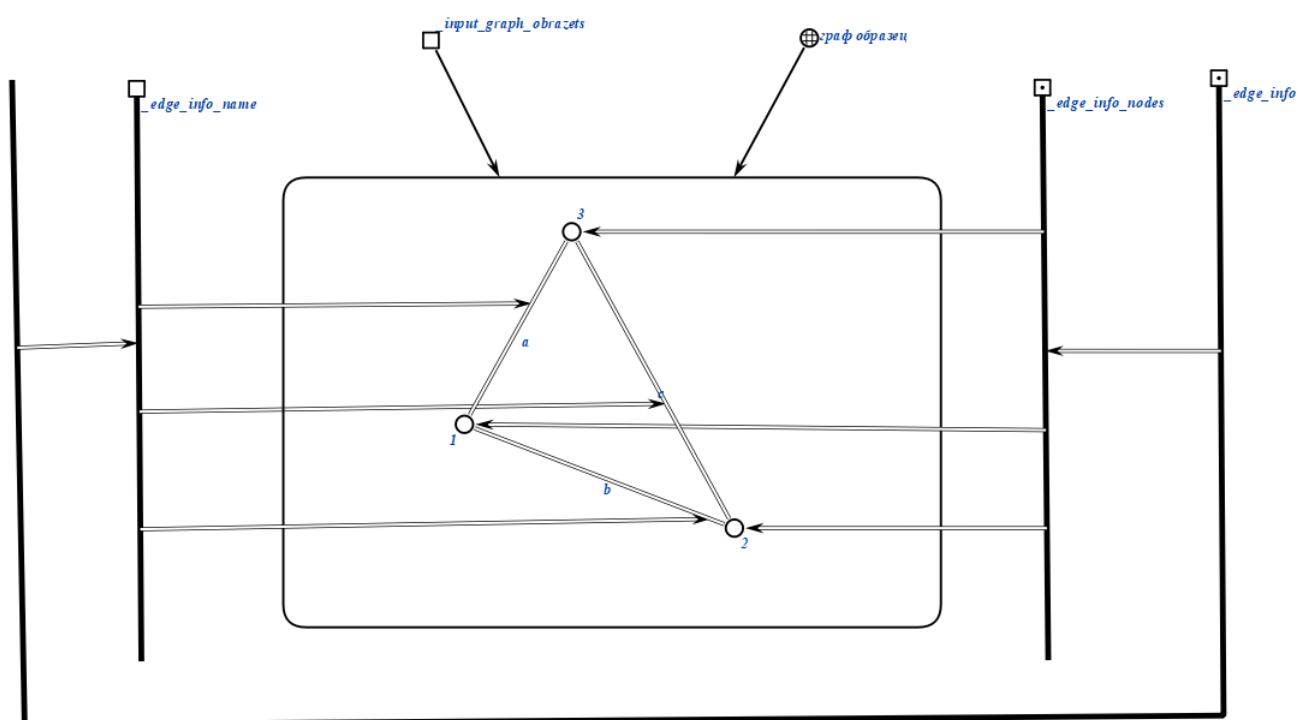


Рисунок 4.4 – Считывание данных из графа-образца в структуру

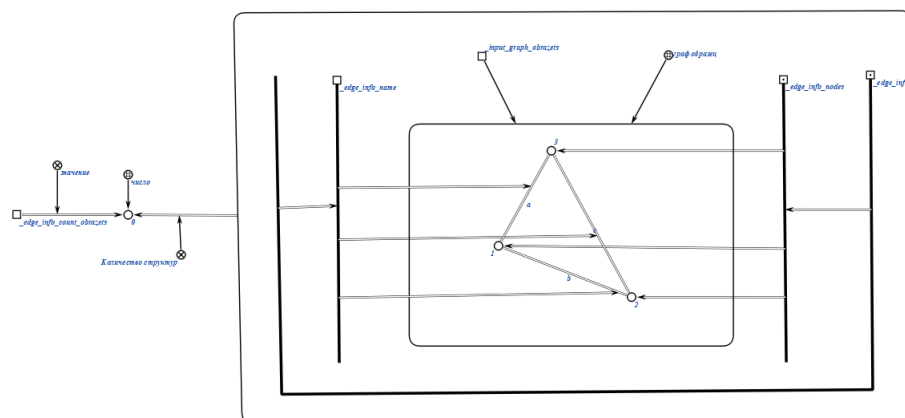


Рисунок 4.5 – Считывание количество структур в переменную

4. Также к дочерней шине, что отвечает за вершины, добавим счётчик, чтобы гарантировать, что в изоморфных графах будет то же самое кол-во структур, что и в графе образце и исключить варианты, когда кол-во структур одинаково, однако кол-во вершин, участвующих в данном графе, не совпадает

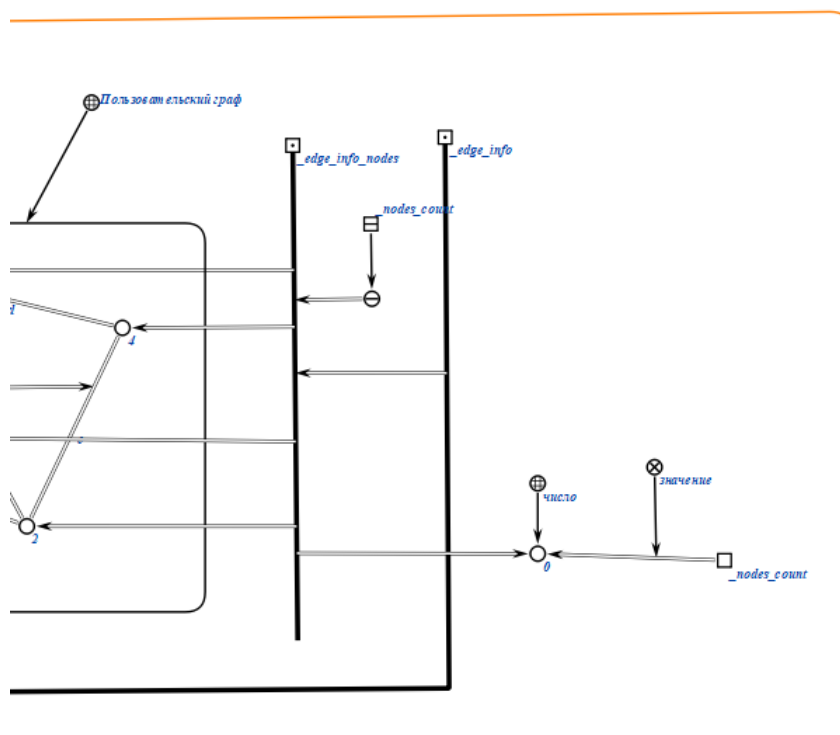


Рисунок 4.6 – Счётчик вершин у исходного графа

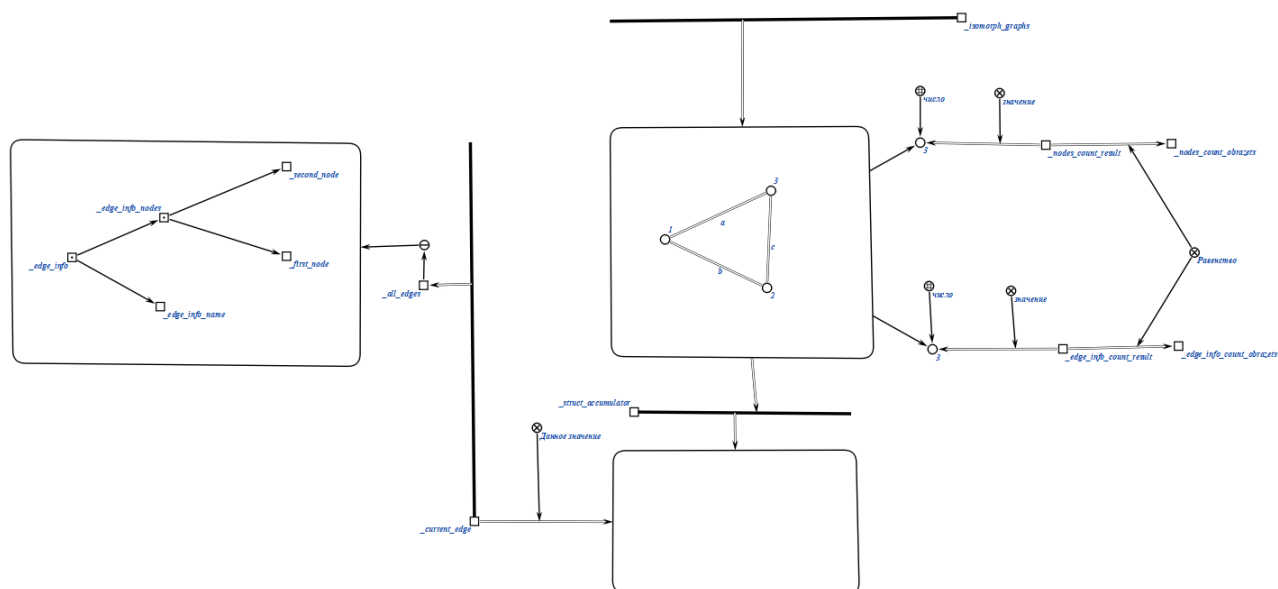


Рисунок 4.13 – Добавление третьей структуры в граф, проверка на кол-во вершин и структур в графе

Если кол-во вершин и структур совпадает, данный граф передаётся по шине в переменную `_isomorph_graphs`, счётчики обнуляются, и цикл начинается по новой.

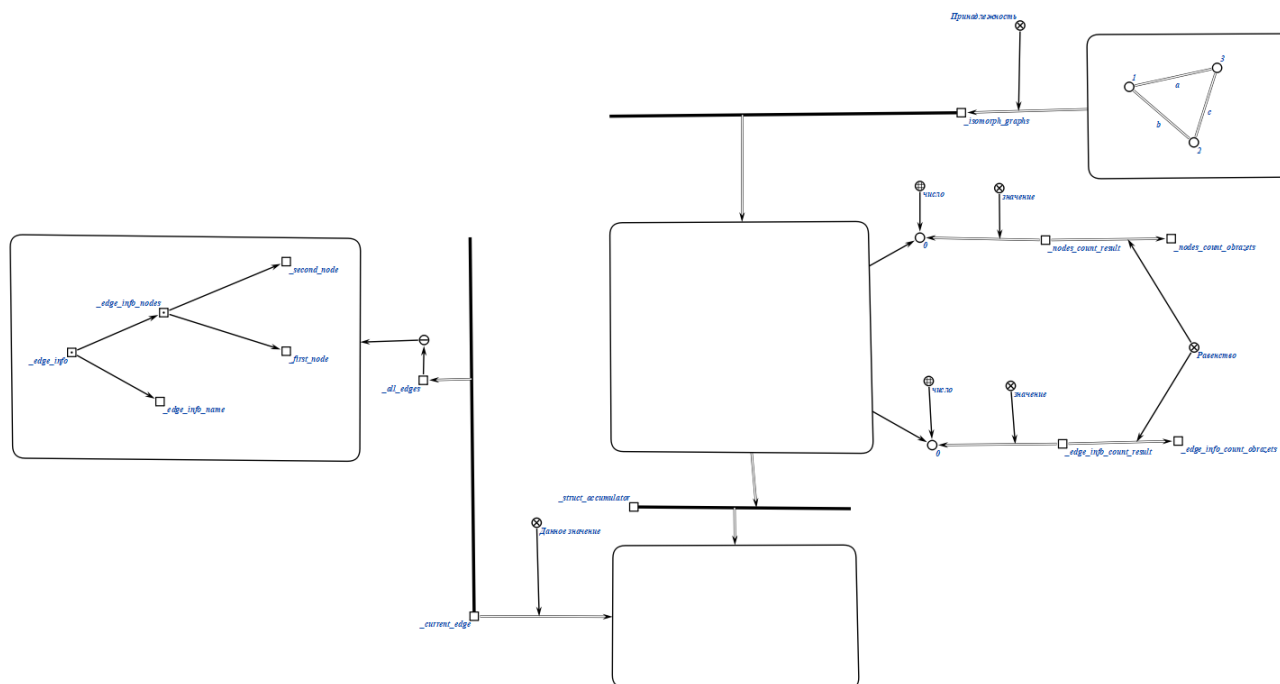


Рисунок 4.14 – Добавление графа в список изоморфных графов

То же самое будет происходить в результате не совпадения, только вместо добавления графа, он будет очищаться

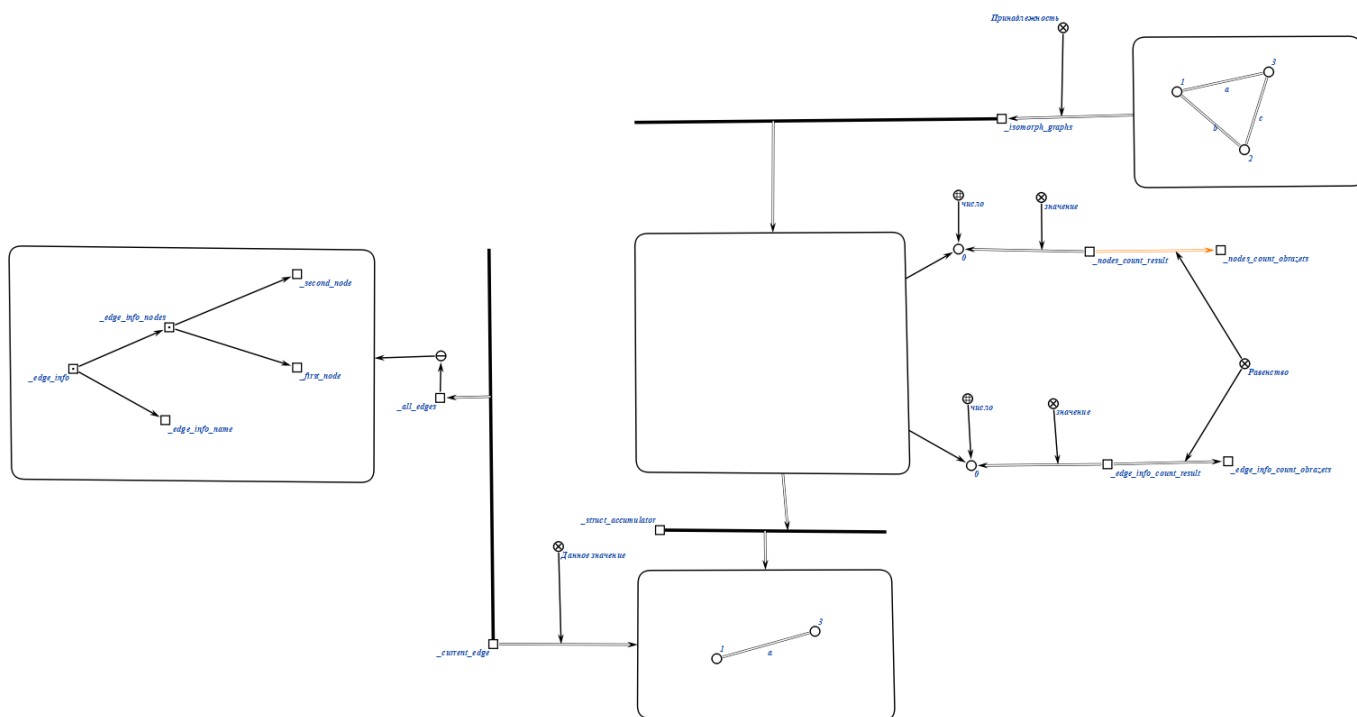


Рисунок 4.15 – Передача первой структуры через шину

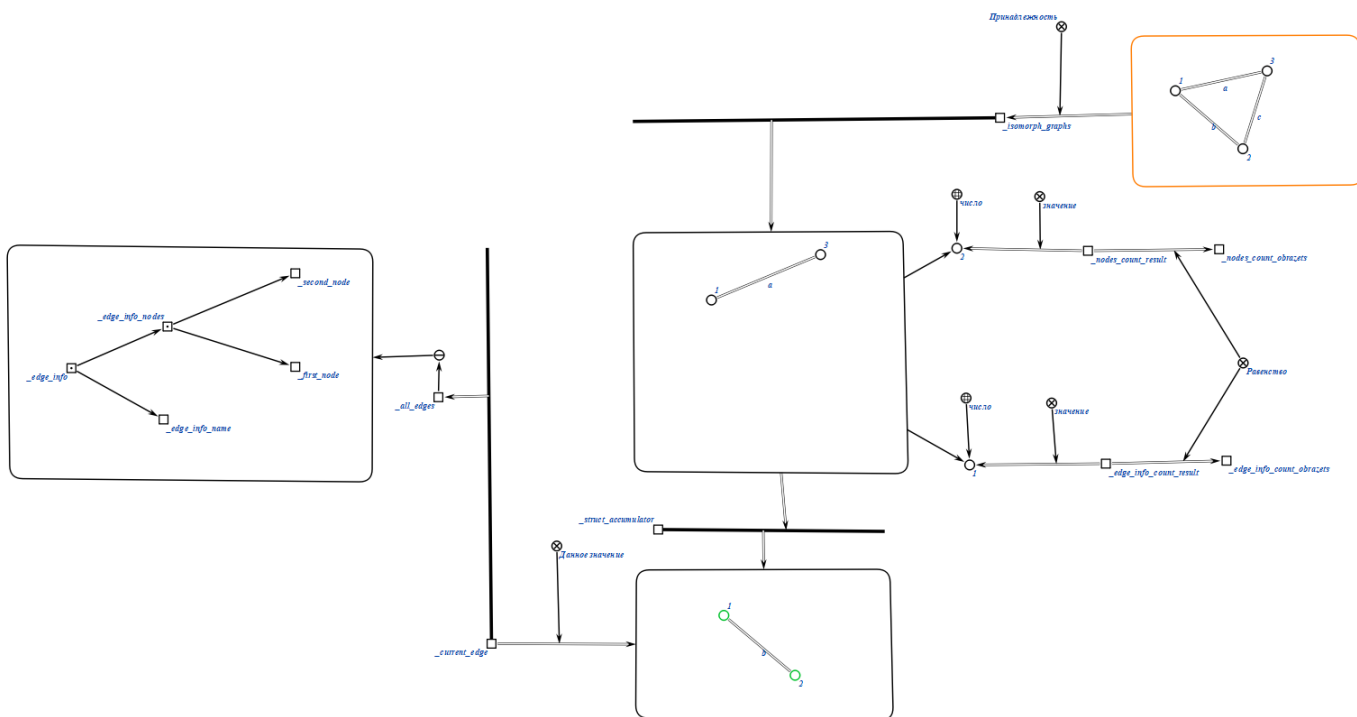


Рисунок 4.16 – Добавление первой структуры в граф, проверка на кол-во вершин и структур в графе

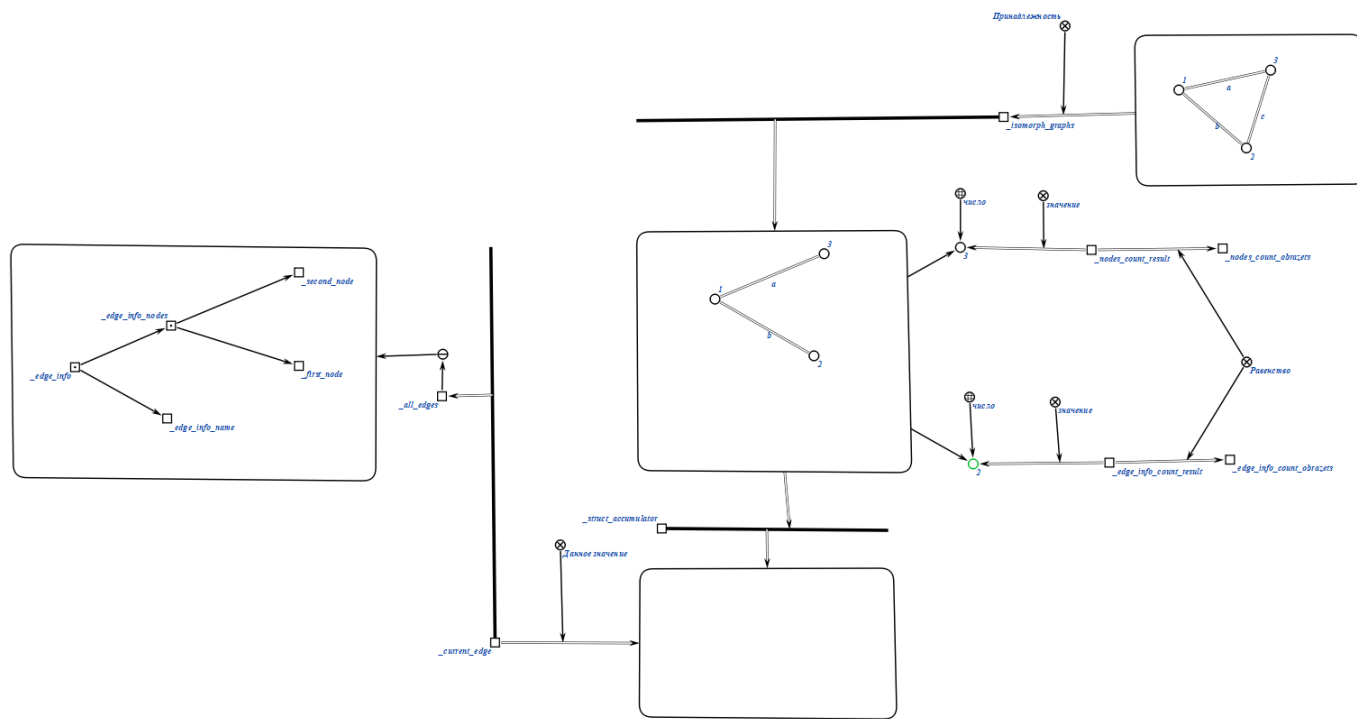


Рисунок 4.17 – Передача второй структуры через шину

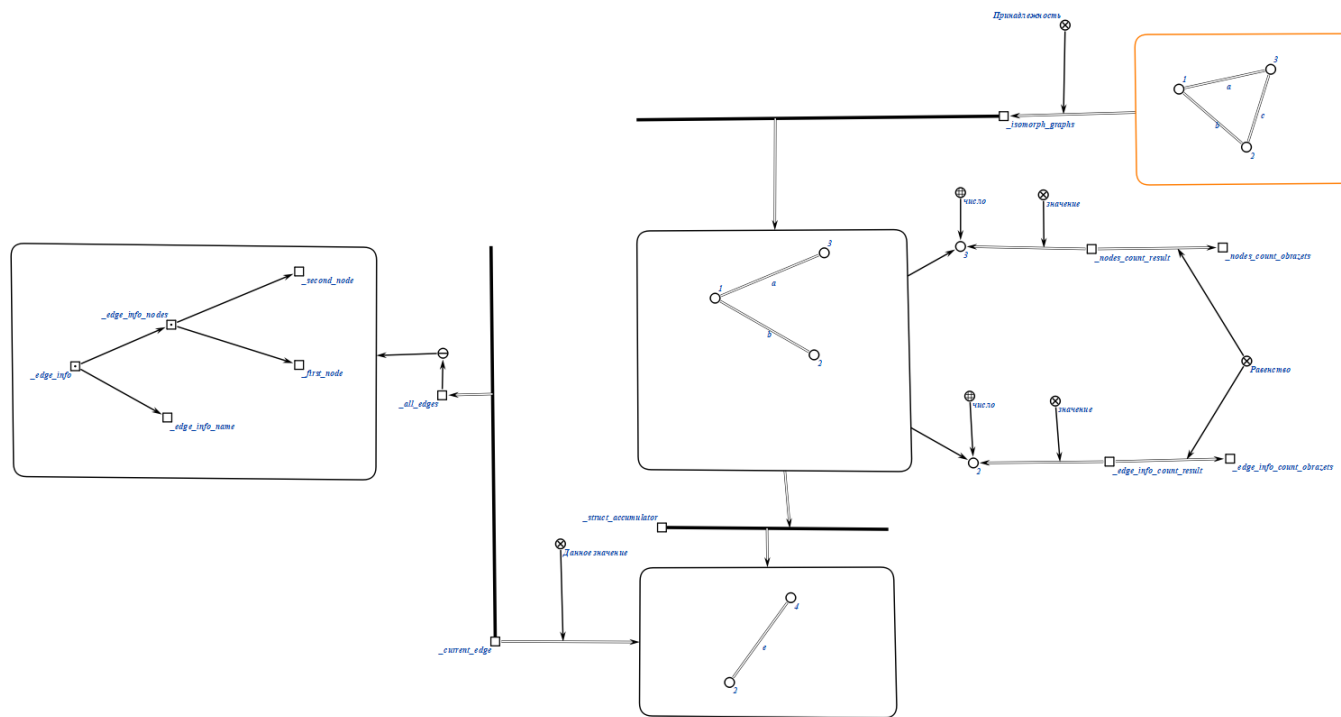


Рисунок 4.18 – Добавление второй структуры в граф, проверка на кол-во вершин и структур в графе

комбинацию структур
В итоге у нас будет только 2 графа

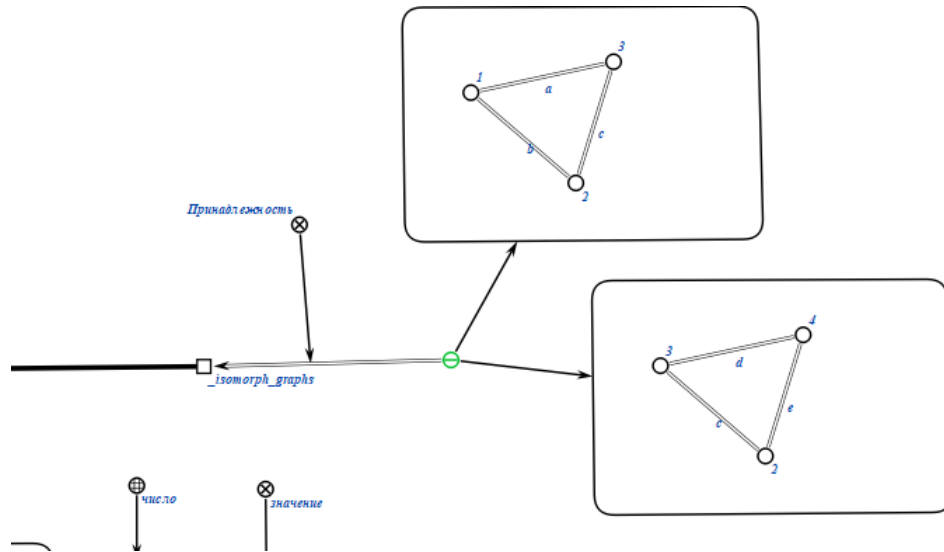


Рисунок 4.21 – Результат алгоритма

6. Вывод ответа: Выводим множество графов, являющихся изоморфными

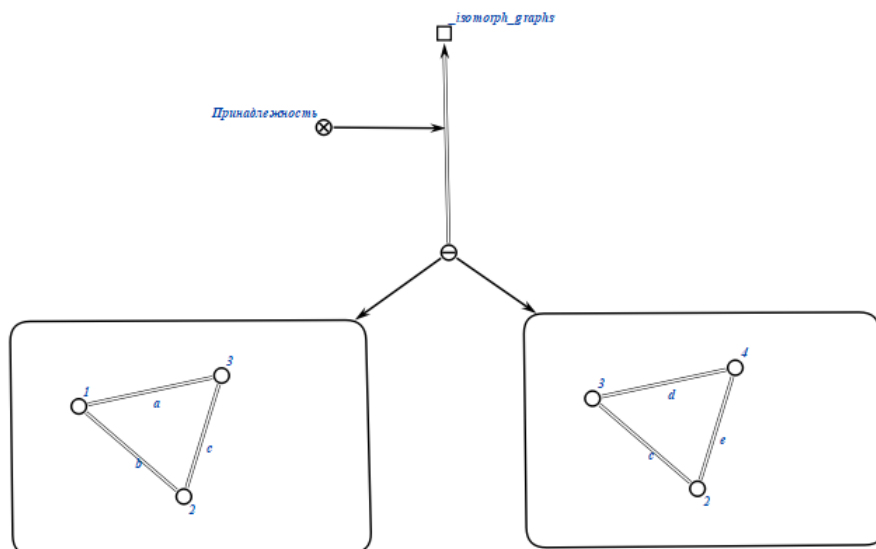


Рисунок 4.22 – Вывод результата

5 ЗАКЛЮЧЕНИЕ

Формализовал алгоритм изоморфных графов

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Кормен, Д. Алгоритмы. Построение и анализ / Д. Кормен. — Вильямс, 2015. — Р. 1328.
- [2] Кузнецов, О. П. Дискретная математика для инженера / О. П. Кузнецов, Г. М. Адельсон-Вельский. — Энергоатомиздат, 1988. — Р. 480.
- [3] Оре, О. Теория графов / О. Оре. — Наука, 1980. — Р. 336.
- [4] Харарри, Ф. Теория графов / Ф. Харарри. — Эдиториал УРСС, 2018. — Р. 304.