

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 формирования множества различных
суграфов неориентированного графа**

Выполнил:

А. В. Клевцевич

Студент группы
821704

Проверил:

Д. В. Шункевич

Минск 2019

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

Задача: Найти различные суграфы неориентированного графа

1 СПИСОК ПОНЯТИЙ

1. Неориентированный граф (абсолютное понятие) – это такой граф, в котором все связи являются ребрами:

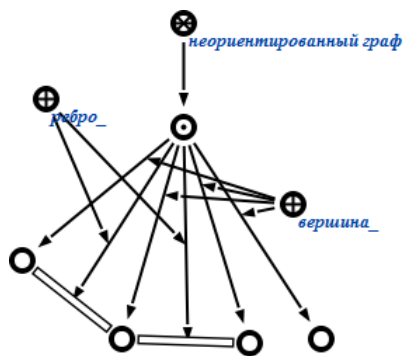


Рисунок 1.1 – Неориентированный граф

2. Суграф - часть графа, имеющая то же множество вершин, что и сам граф.

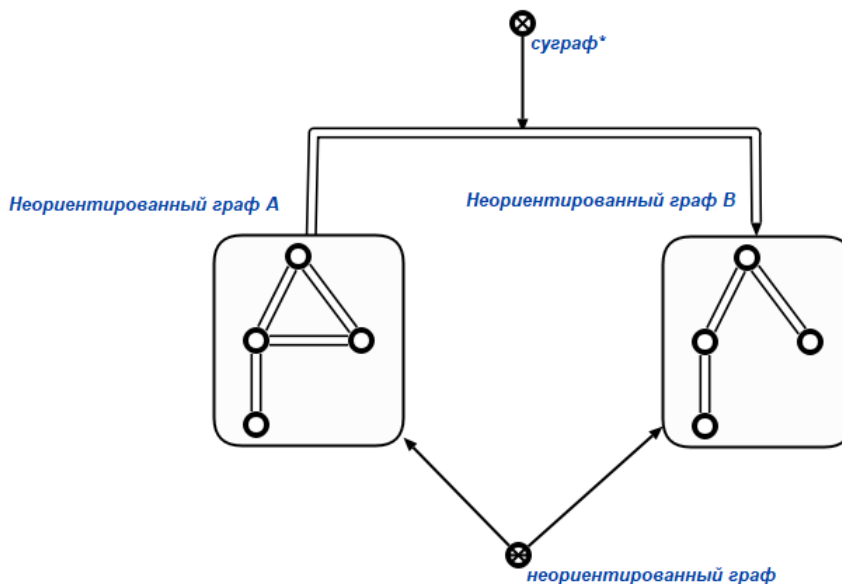


Рисунок 1.2 – Граф В является суграфом графа А

2 ТЕСТОВЫЕ ПРИМЕРЫ

Во всех тестах графы будут приведены в сокращенной форме со скрытыми ролями элементов графа.

2.1 Тест 1

Вход:

Необходимо найти различные суграфы графа A .

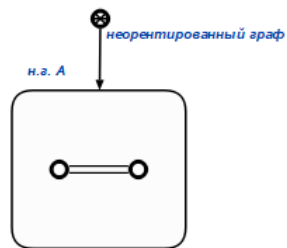


Рисунок 2.1 – Вход теста 1

Выход:

Будет найдено 2 различных суграфа B , B графа A :

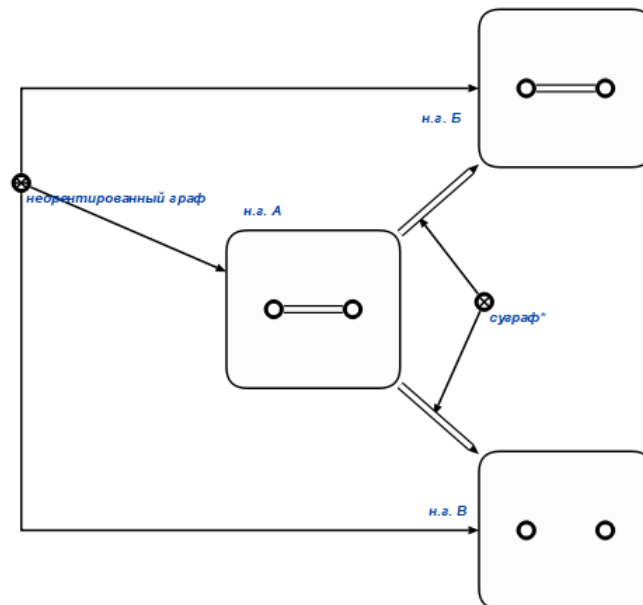


Рисунок 2.2 – Выход теста 1

2.2 Тест 2

Вход:

Необходимо найти различные суграфы графа A .

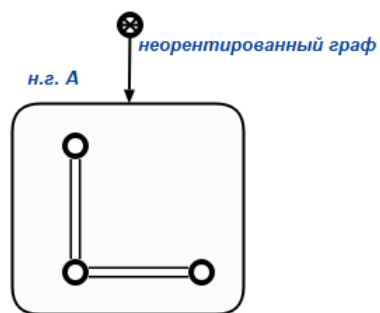


Рисунок 2.3 – Вход теста 2

Выход:

Будет найдено 4 различных суграфа B , B , Γ , Δ графа A :

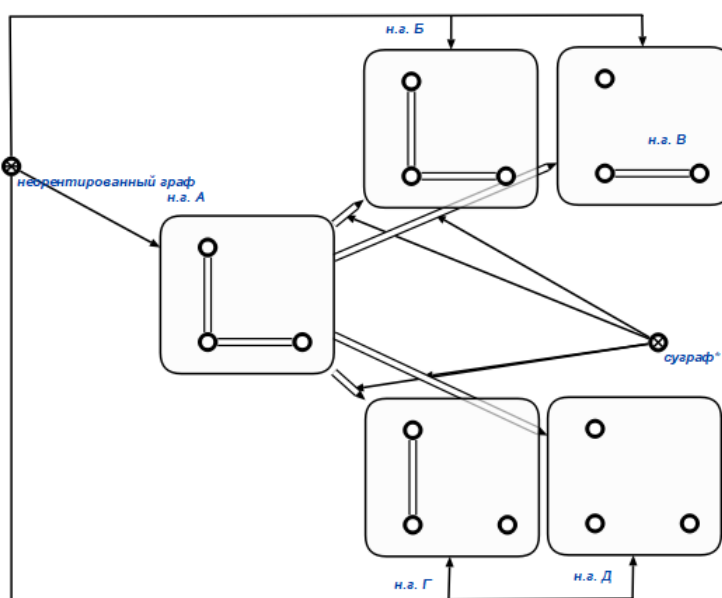


Рисунок 2.4 – Выход теста 2

2.3 Тест 3

Вход:

Необходимо найти различные суграфы графа A .

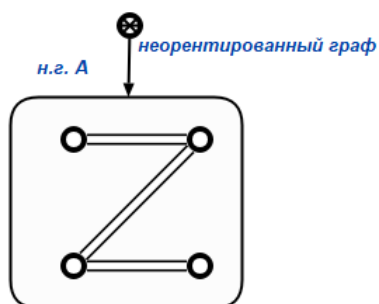


Рисунок 2.5 – Вход теста 3

Выход:

Будет найдено 8 различных суграфа B , B , Γ , Δ , E , \mathcal{J} , \mathcal{Z} , \mathcal{I} графа A :

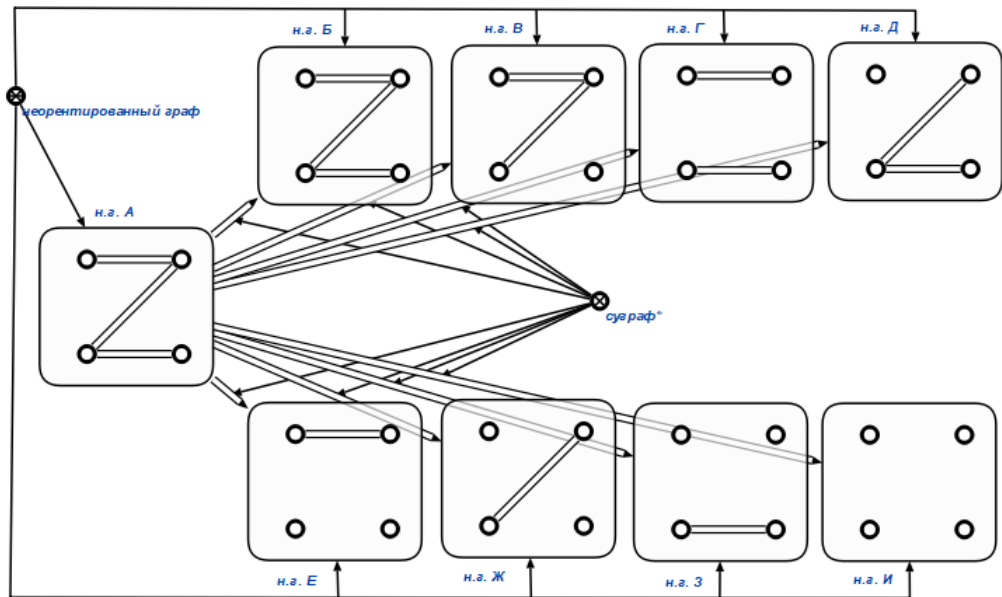


Рисунок 2.6 – Выход теста 3

2.4 Тест 4

Вход:

Необходимо найти различные суграфы графа A .

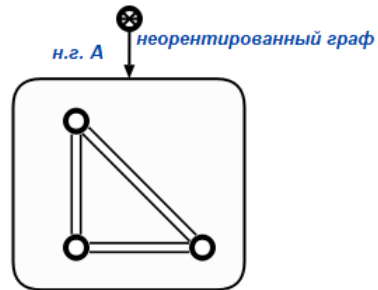


Рисунок 2.7 – Вход теста 4

Выход:

Будет найдено 8 различных суграфа B , B , Γ , Δ , E , \mathcal{J} , \mathcal{Z} , \mathcal{I} графа A :

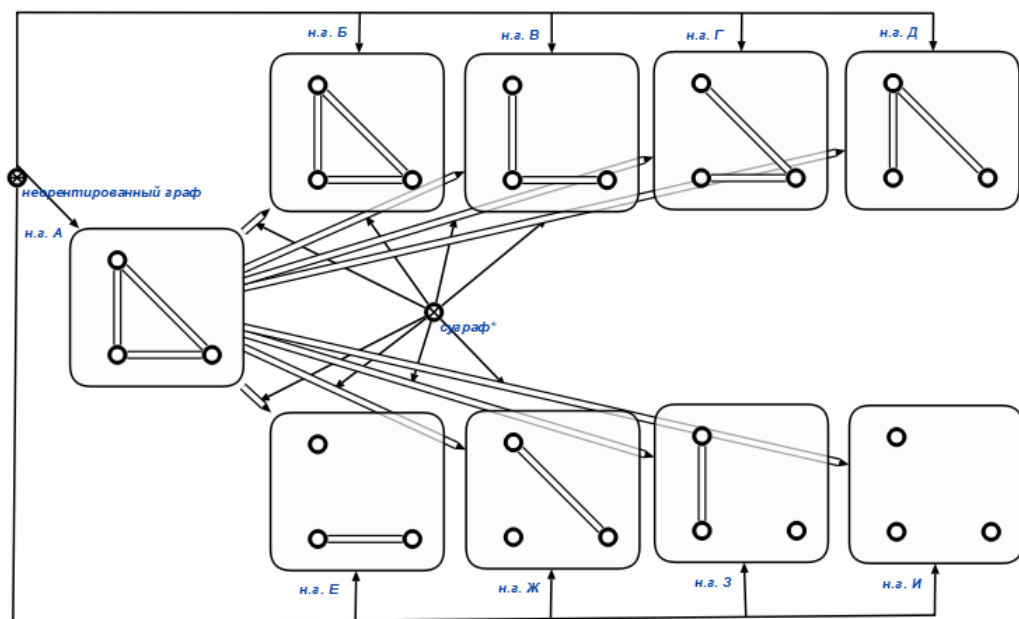


Рисунок 2.8 – Выход теста 4

2.5 Тест 5

Вход:

Необходимо найти различные суграфы графа А.

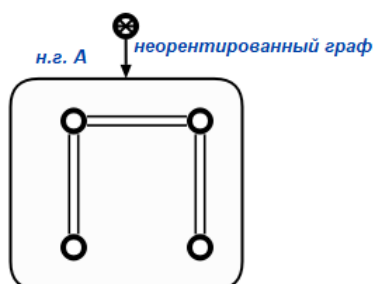


Рисунок 2.9 – Вход теста 5

Выход:

Будет найдено 8 различных суграфа Б, В, Г, Д, Е, Ж, З, И графа А:

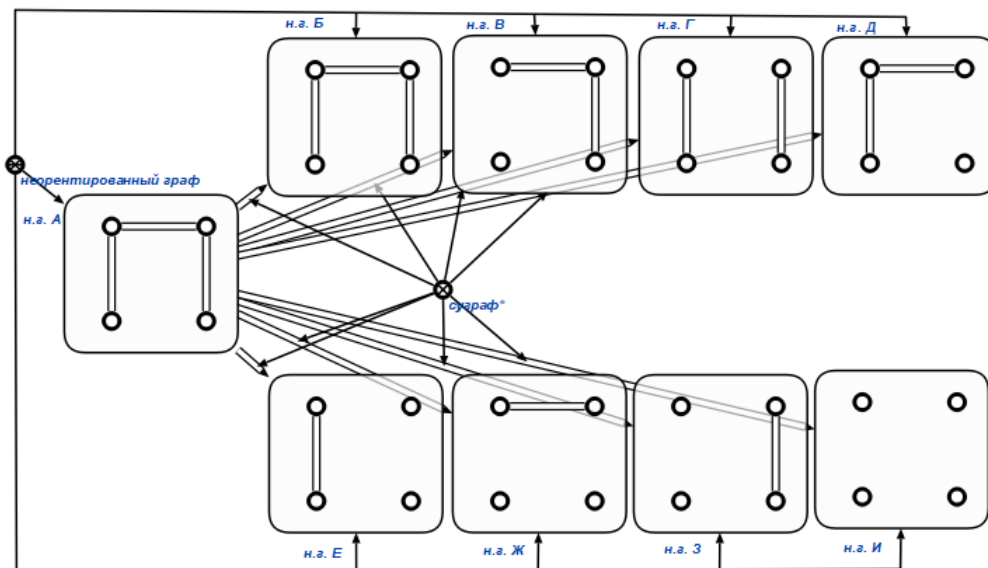


Рисунок 2.10 – Выход теста 5

3 АЛГОРИТМ ПОИСКА РАЗЛИЧНЫХ СУГРАФОВ ГРАФА

3.1 Описание алгоритма

1. Формируем множество суграфов. Добавляем исходный граф во множество суграфов.
2. $i = 1$.
3. В новый граф $_g$ присваиваем i -ый элемент множества.
4. Выбираем любое ребро в качестве начального. В $_j$ присваиваем начальное ребро.
5. Создаем множество непроверенных ребер, добавляем в него все ребра, кроме начального.
6. Удаляем j -ое ребро из $_g$
7. Если такого графа нет в множестве суграфов, то добавляем $_g$ в множество
8. Возвращаем j -ое ребро в $_g$.
9. Присваиваем $_j$ любое ребро из множества непроверенных. Удаляем j -ое ребро из множества непроверенных ребер.
10. Возвращаемся к 4 пункту, пока множество непроверенных ребер не пустое.
11. Увеличиваем i на 1.
12. Возвращаемся к 3 пункту, пока i не равно числу элементов множества.
13. Получаем множество суграфов данного графа.
14. Завершить алгоритм.

3.2 Переменные

Используемые переменные:

- `_g` – промежуточный граф для поиска суграфов.
- `_not_checked_edges` – множество непроверенных ребер
- `_i` – порядковый номер графа во множестве суграфов.
- `_j` – текущее ребро

3.3 Пример выполнения алгоритма в сс-памяти

1. Задание входного графа, начальной точки для работы алгоритма.

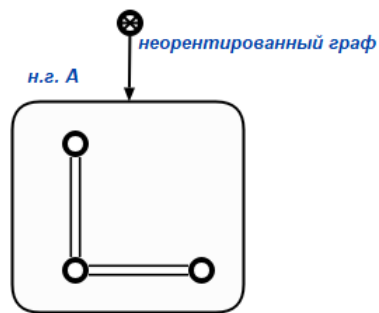


Рисунок 3.1 – Входной граф

2. Формируем множество суграфов. Добавляем исходный граф во множество суграфов.

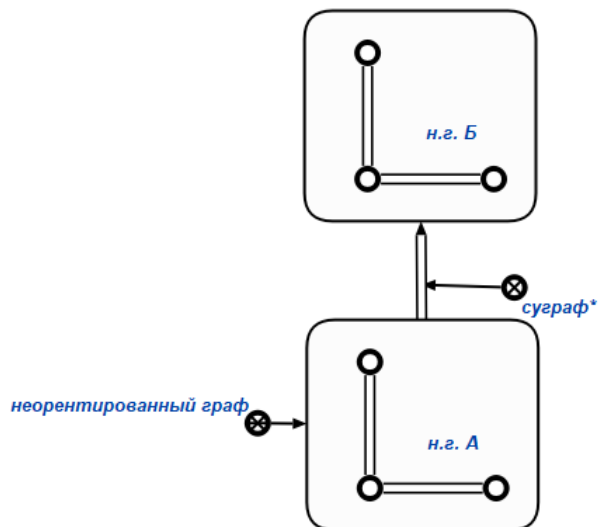


Рисунок 3.2 – Выполнение алгоритма

3. `_i = 1`.

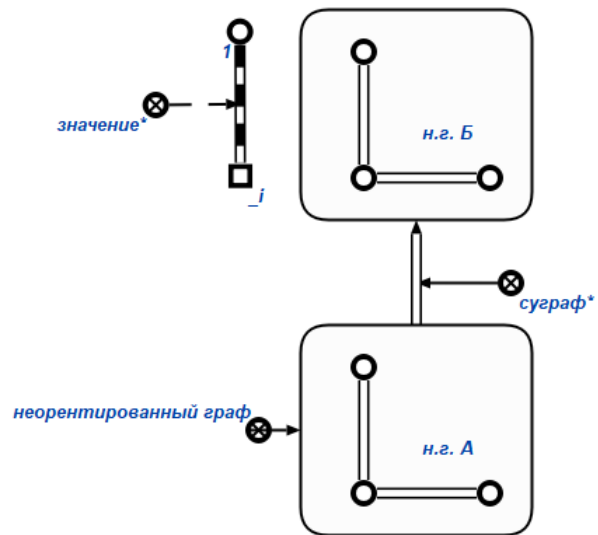


Рисунок 3.3 – Выполнение алгоритма

4. В новый граф $_g$ присваиваем $_i$ -ый элемент множества.

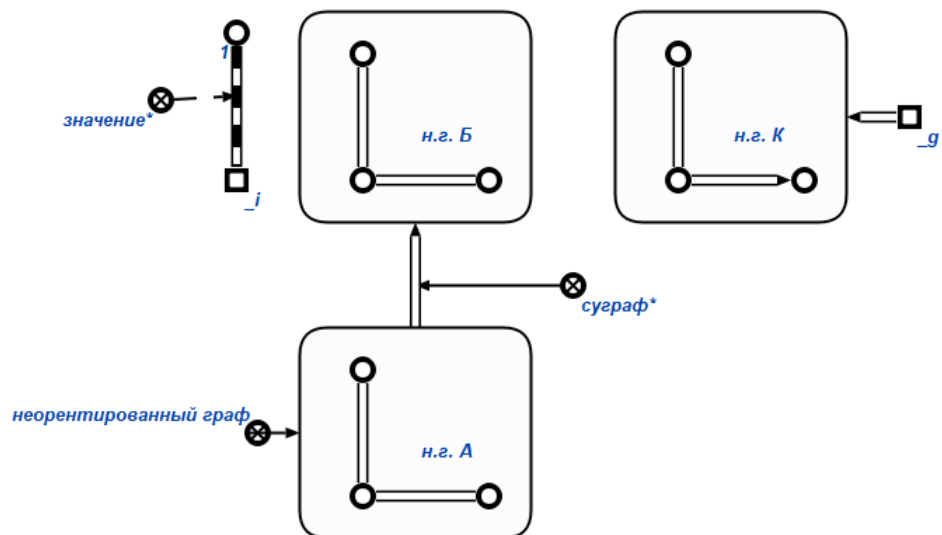


Рисунок 3.4 – Выполнение алгоритма

5. Выбираем любое ребро в качестве начального. В $_j$ присваиваем начальное ребро.

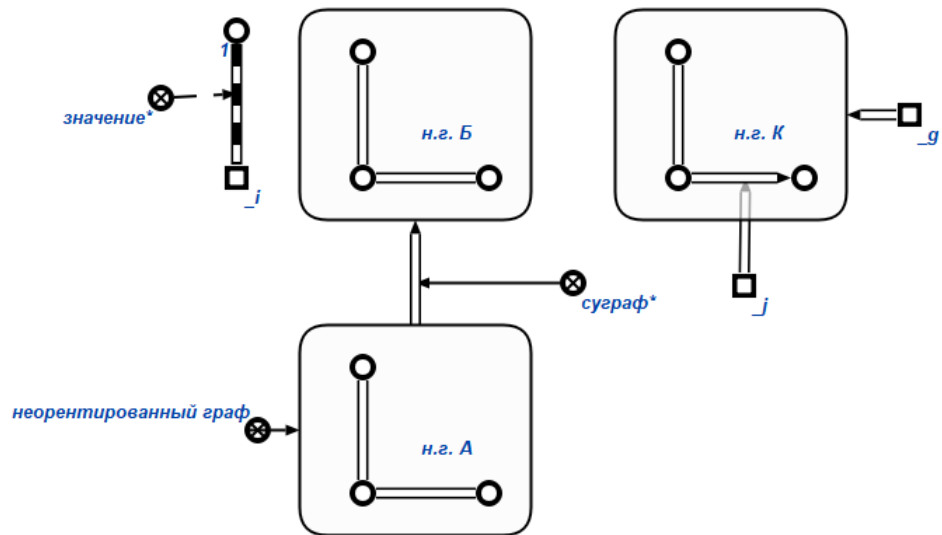


Рисунок 3.5 – Выполнение алгоритма

6. Создаем множество непроверенных ребер, добавляем в него все ребра, кроме начального.

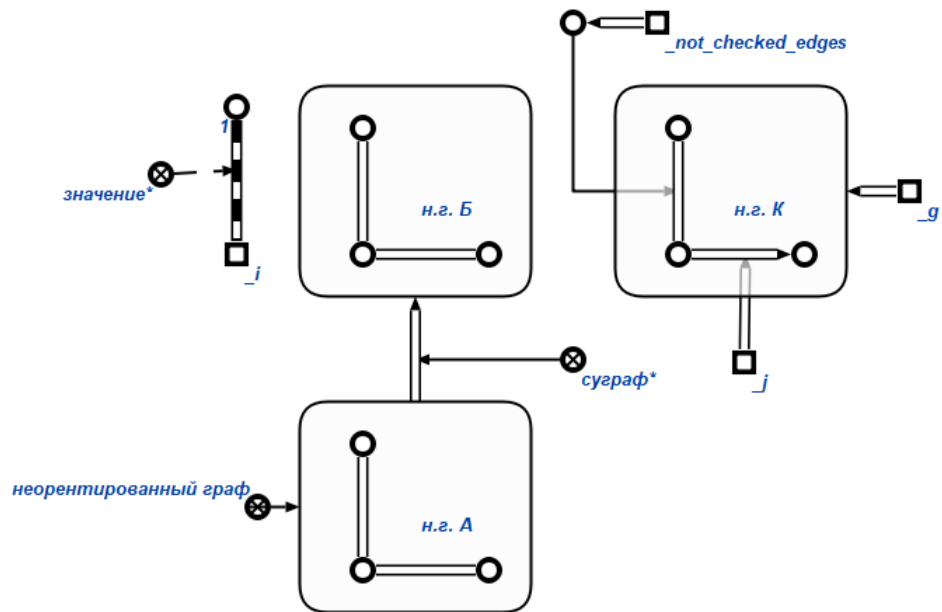


Рисунок 3.6 – Выполнение алгоритма

Переменная `_not_checked_edges` получит в качестве значения множество непроверенных ребер обрабатываемого графа.

7. Удаляем `_j`-ое ребро из `_g`.

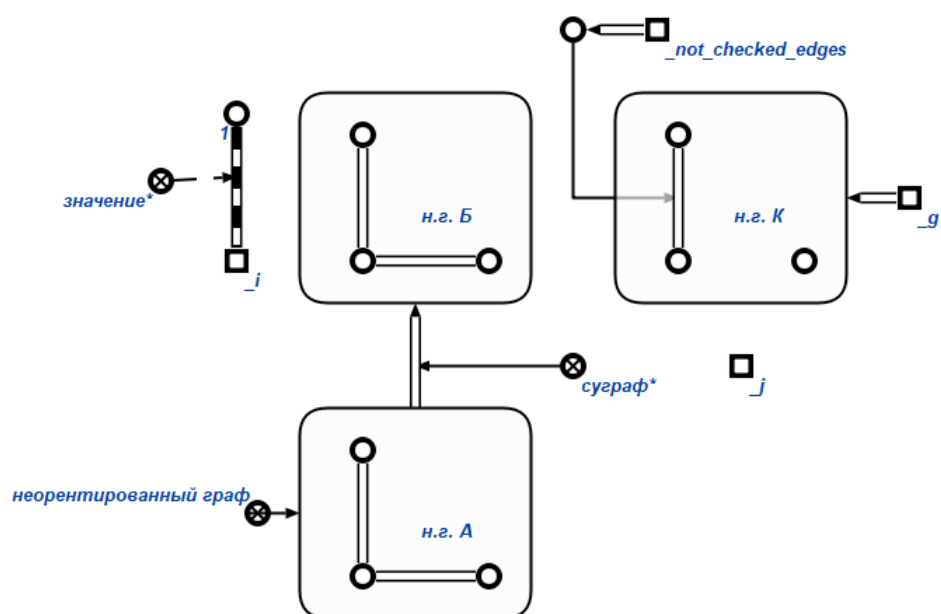


Рисунок 3.7 – Выполнение алгоритма

8. Добавляем $_g$ в множество суграфов.

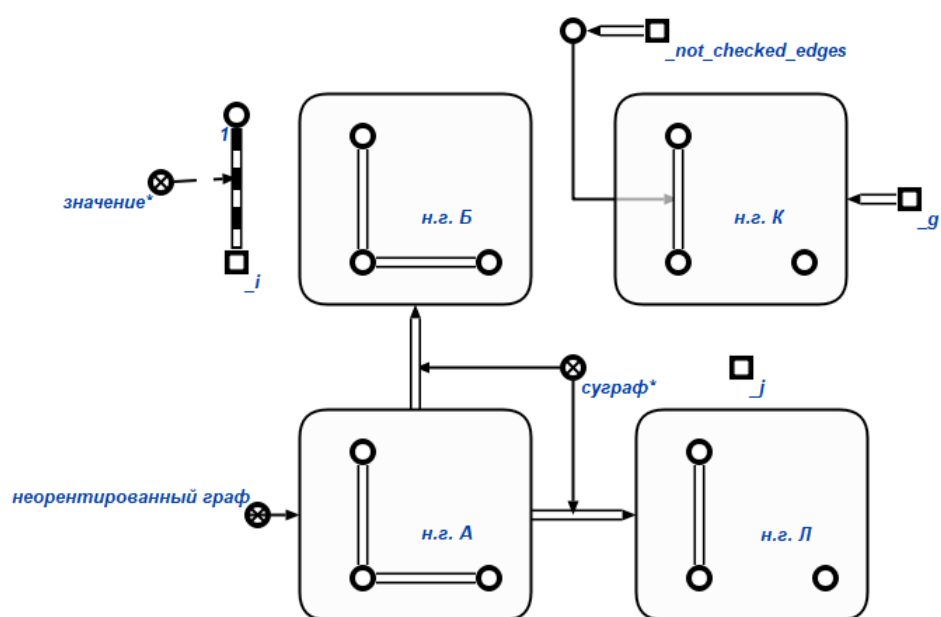


Рисунок 3.8 – Выполнение алгоритма

9. Возвращаем j -ое ребро в $_g$.

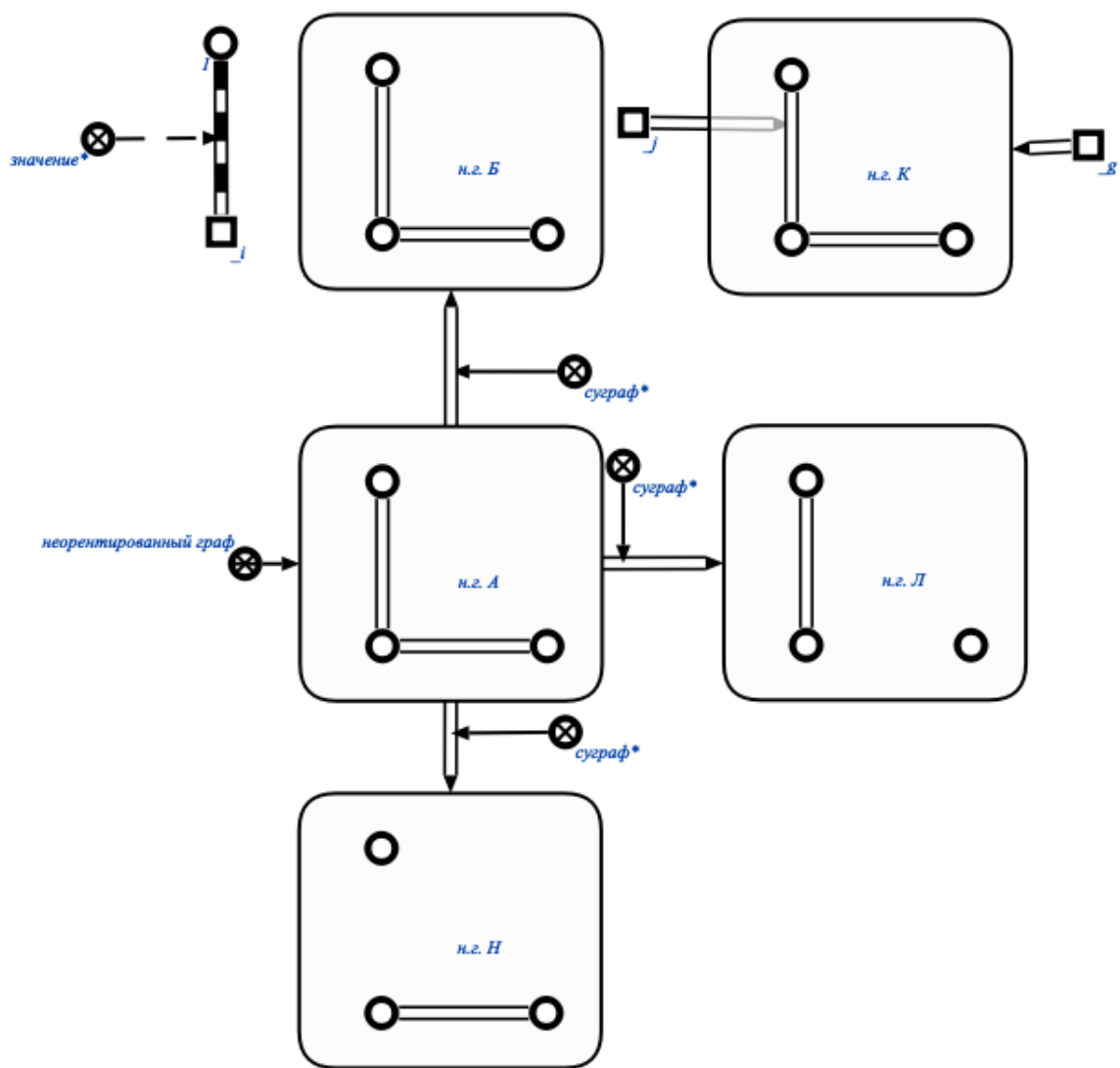


Рисунок 3.11 – Выполнение алгоритма

12. Увеличиваем $_i$ на 1.

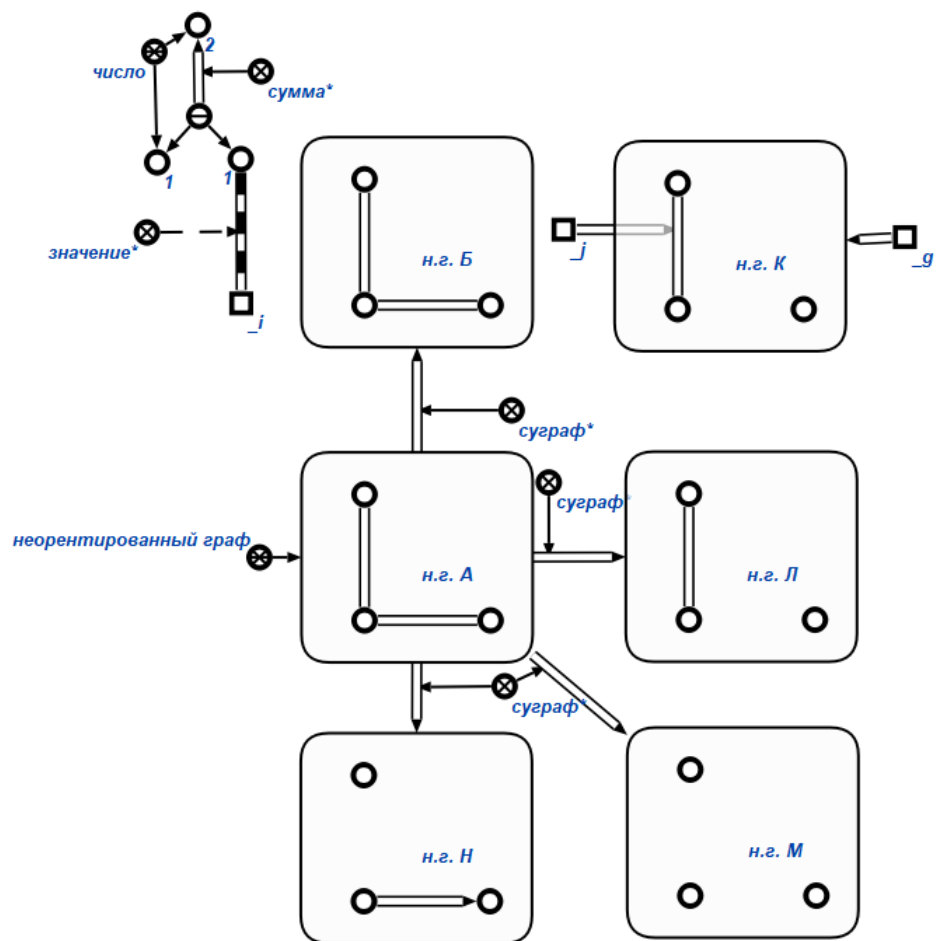


Рисунок 3.13 – Выполнение алгоритма

14. Вывод:

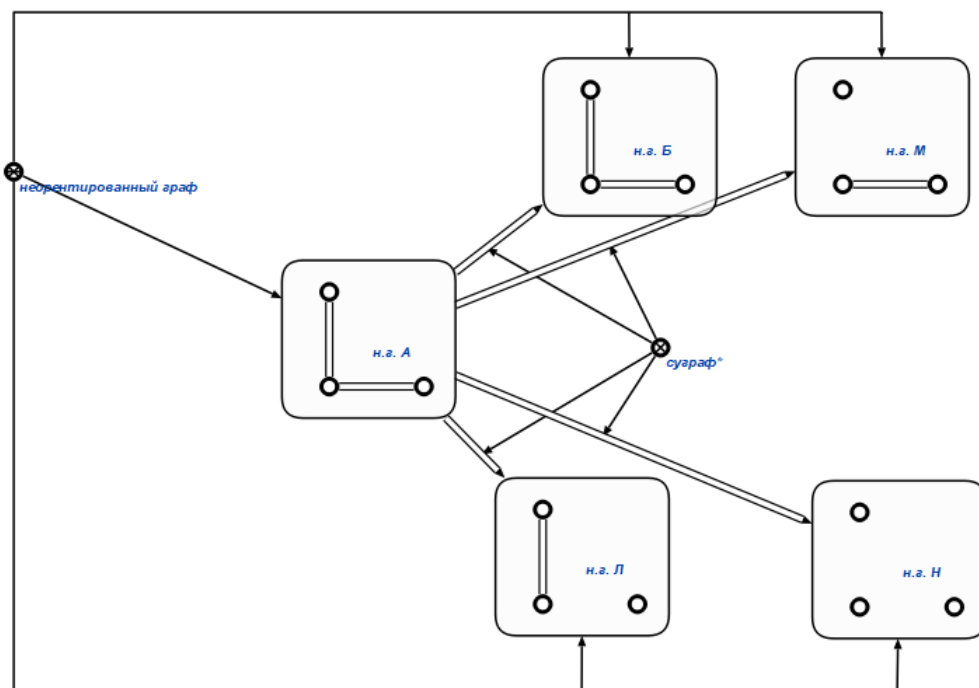


Рисунок 3.14 – Результат выполнения алгоритма

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.