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 «Представление и обработка информации в интеллектуальных системах»

на тему

Найти реберный граф для неориентированного графа

Выполнила: В. А. Казаченко

Студент группы
321702

Проверил: Н. В. Малиновская

Минск 2024

Содержание

1 Введение

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

Задача: Найти реберный граф для неориентированного графа.

2 Список понятий

1. ***Граф*** (Рис.1) (абсолютное понятие) - совокупность непустого множества вершин и наборов пар вершин (связей между вершинами).

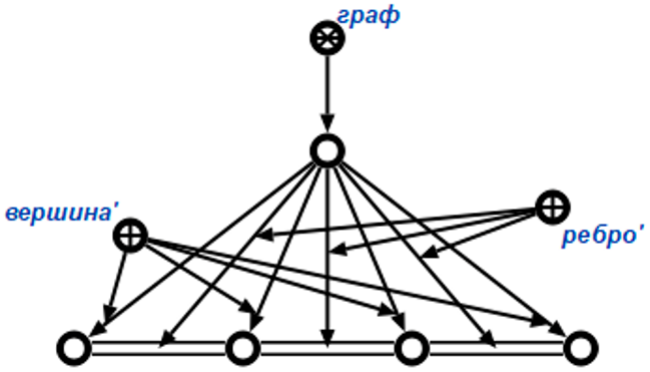

Рис. 1: Граф

2. ***Неориентированный граф*** (Рис.2) (абсолютное понятие) – граф, в котором все связки-ребра.

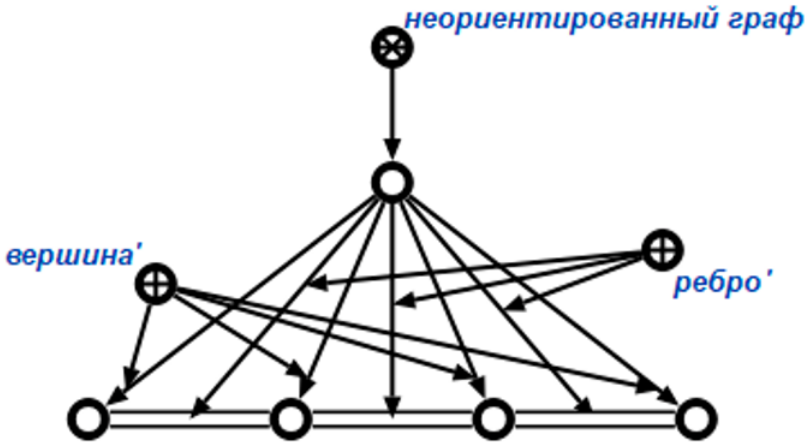

Рис. 2: Неориентированный граф

3. ***Взвешенный граф*** (Рис.3) (абсолютное понятие) – граф, каждому ребру которого поставлено в соответствие некое значение (вес ребра).

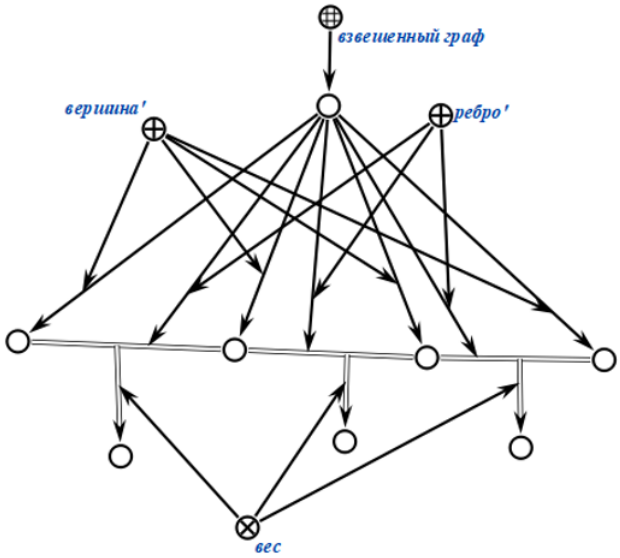

Рис. 3: Взвешенный граф

4. ***Связный граф*** (Рис.4) (абсолютное понятие) – граф, содержащий только одну компоненту связности.

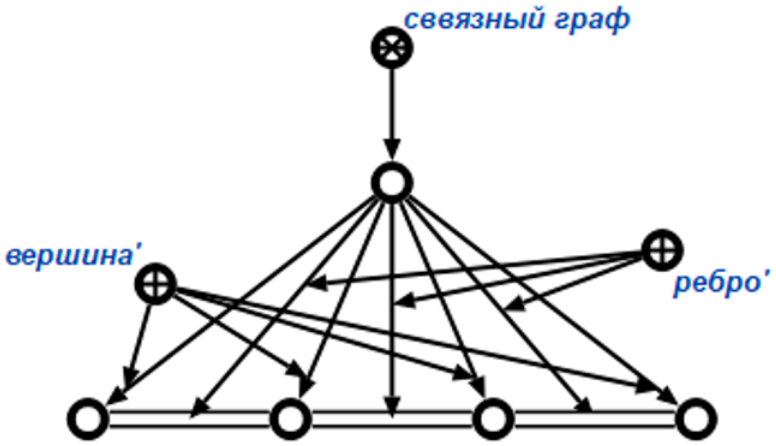

Рис. 4: Связный граф

3 Тестовые примеры

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

3.1 Тест 1

Вход: Необходимо найти реберный граф для неориентированного графа.

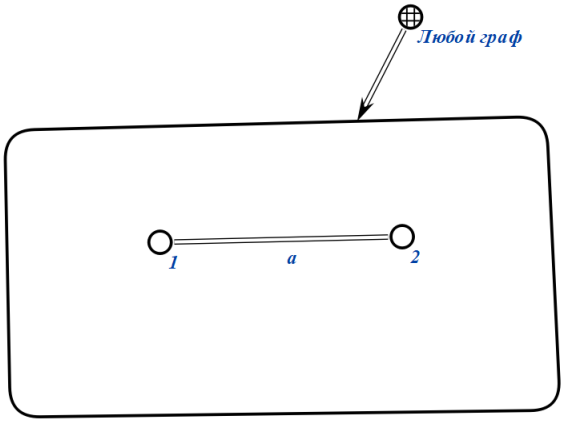

Рис. 5: Вход теста 1

Выход:

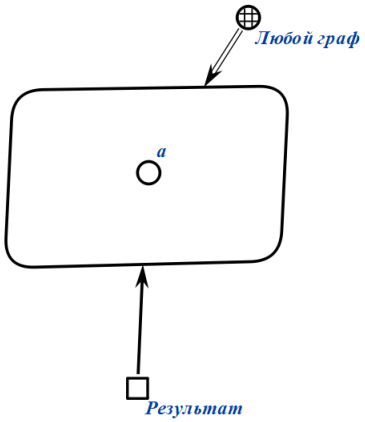

Рис. 6: Выход теста 1

3.2 Тест 2

Вход: Необходимо найти реберный граф для неориентированного графа.

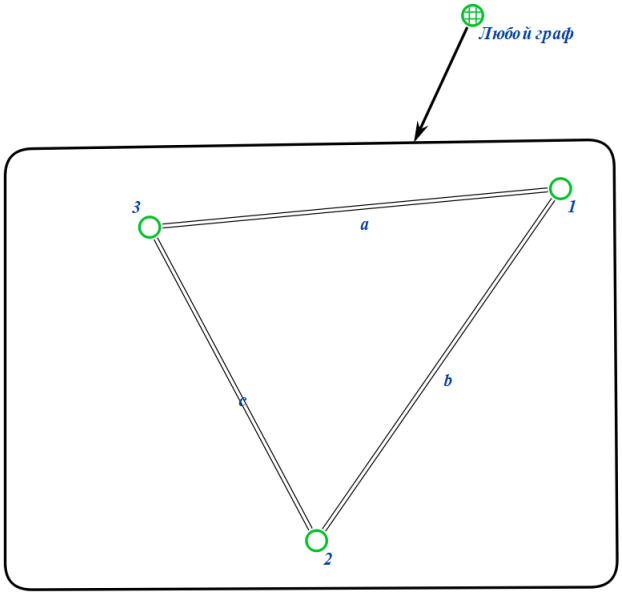

Рис. 7: Вход теста 2

Выход:

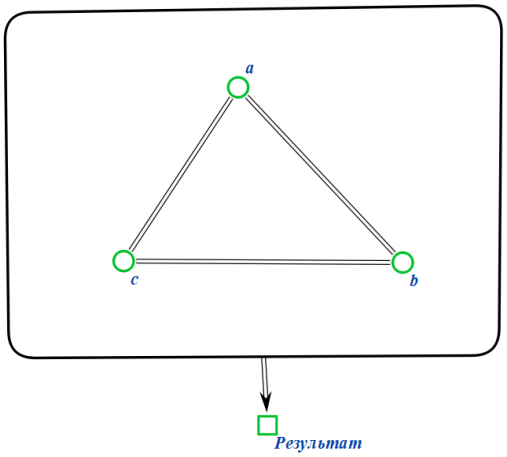

Рис. 8: Выход теста 2

3.3 Тест 3

Вход: Необходимо найти реберный граф для неориентированного графа.

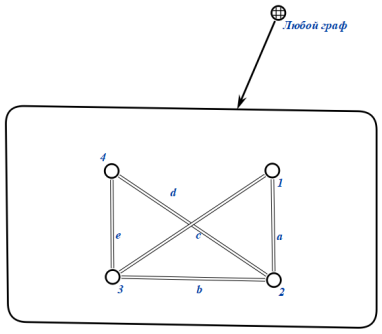

Рис. 9: Вход теста 3

Выход:

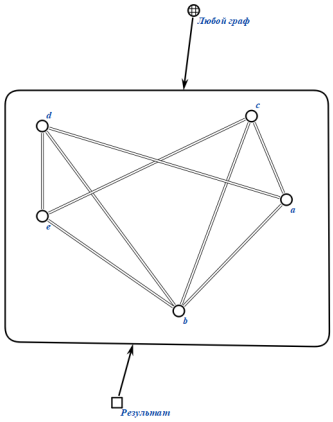

Рис. 10: Выход теста 3

3.4 Тест 4

Вход: Необходимо найти реберный граф для неориентированного графа.

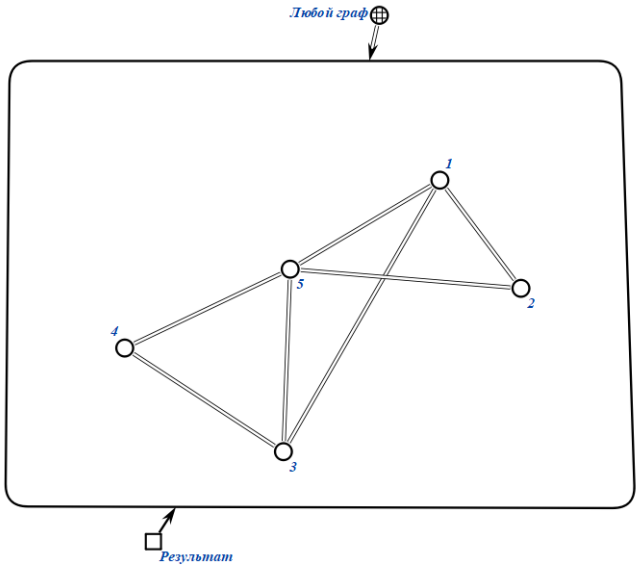

Рис. 11: Вход теста 4

Выход:

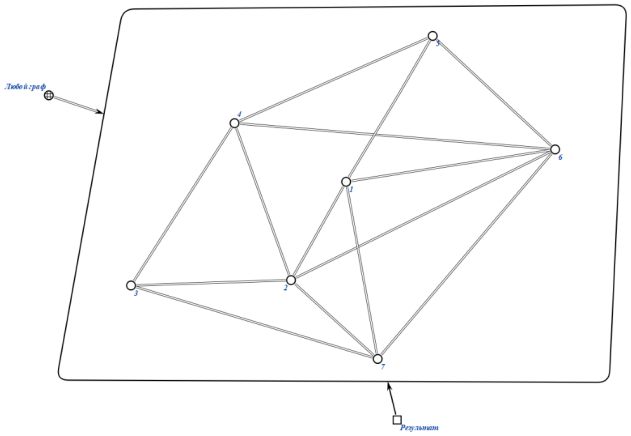

Рис. 12: Выход теста 4

3.5 Тест 5

Вход: Необходимо найти реберный граф для неориентированного графа.

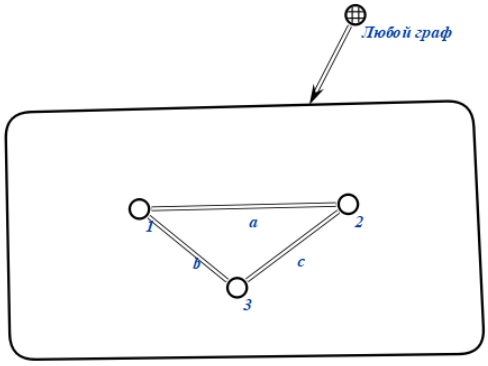

Рис. 13: Вход теста 5

Выход:

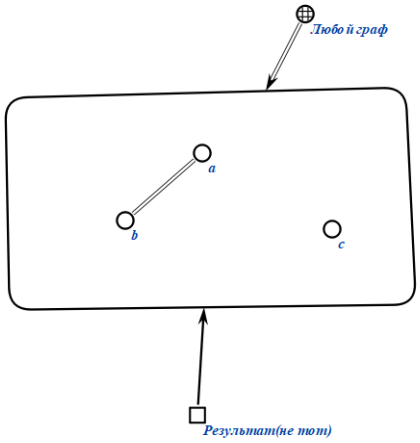

Рис. 14: Выход теста 5

4 Алгоритм

4.1 Краткое описание алгоритма:

1. Первый элемент первого уровня содержит список
2. Создаем счетчик для отслеживания количества ребер, в последующем этот счетчик будет увеличиваться;
3. Получем количество вершин и ребер;
4. Создаем обьекты для перебора всех вершин и ребер, изначально они установлена на первые вершину и ребро
5. Счетчик пока остается на 0
6. Создаем объект для нахождения всех смежных ребер для кадого ребра
7. Создаем счетчик смежных ребер для каждого ребра
8. Начинаем проверять первое ребро
9. Проверяем до тех пор пока не найдем две вершины
10. Проверяем второе ребро и все оставшиеся с каждой вершиной
11. Счетчик при это будет получать новое значение в зависимости от ребер каждый раз как будет проверено ребро
12. После того, как мы проверили все ребра и нашли их количество, начинаем строить новый граф по следующим условиям:

    12.1) Количество вершин нового графа должно быть равно количеству ребер изначального графа.

    12.2) Граф неориетированный

13. построим новый граф на примере этого.
14. Завершение алгоритма.

5 Пример выполнения алгоритма в sc-памяти для графа из теста 4:

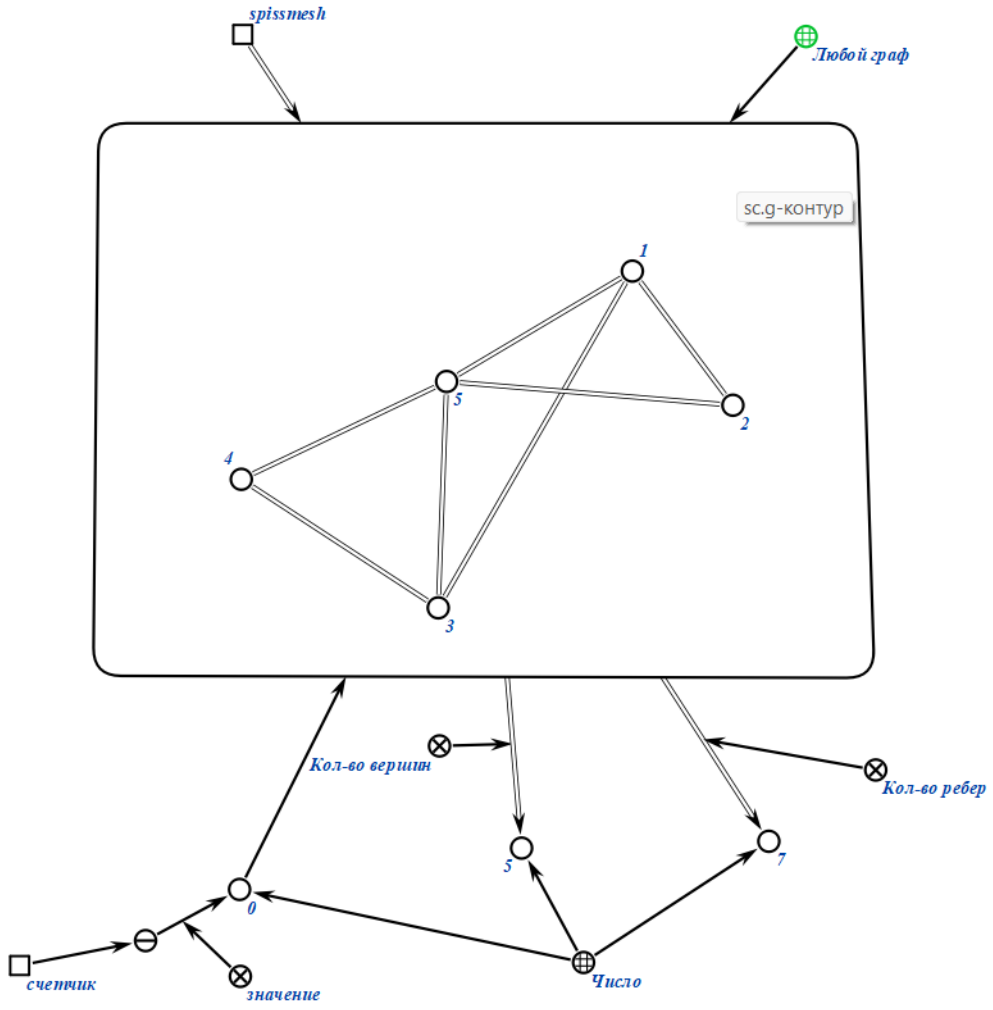

Рис. 15: Вход теста 5

1. ***spissmesh*** получит в качестве значения sc-узел неориентированного графа;
2. Создаем счетчик для отслеживания количества ребер, в последующем этот счетчик будет увеличиваться
3. получем количество вершин и ребер;

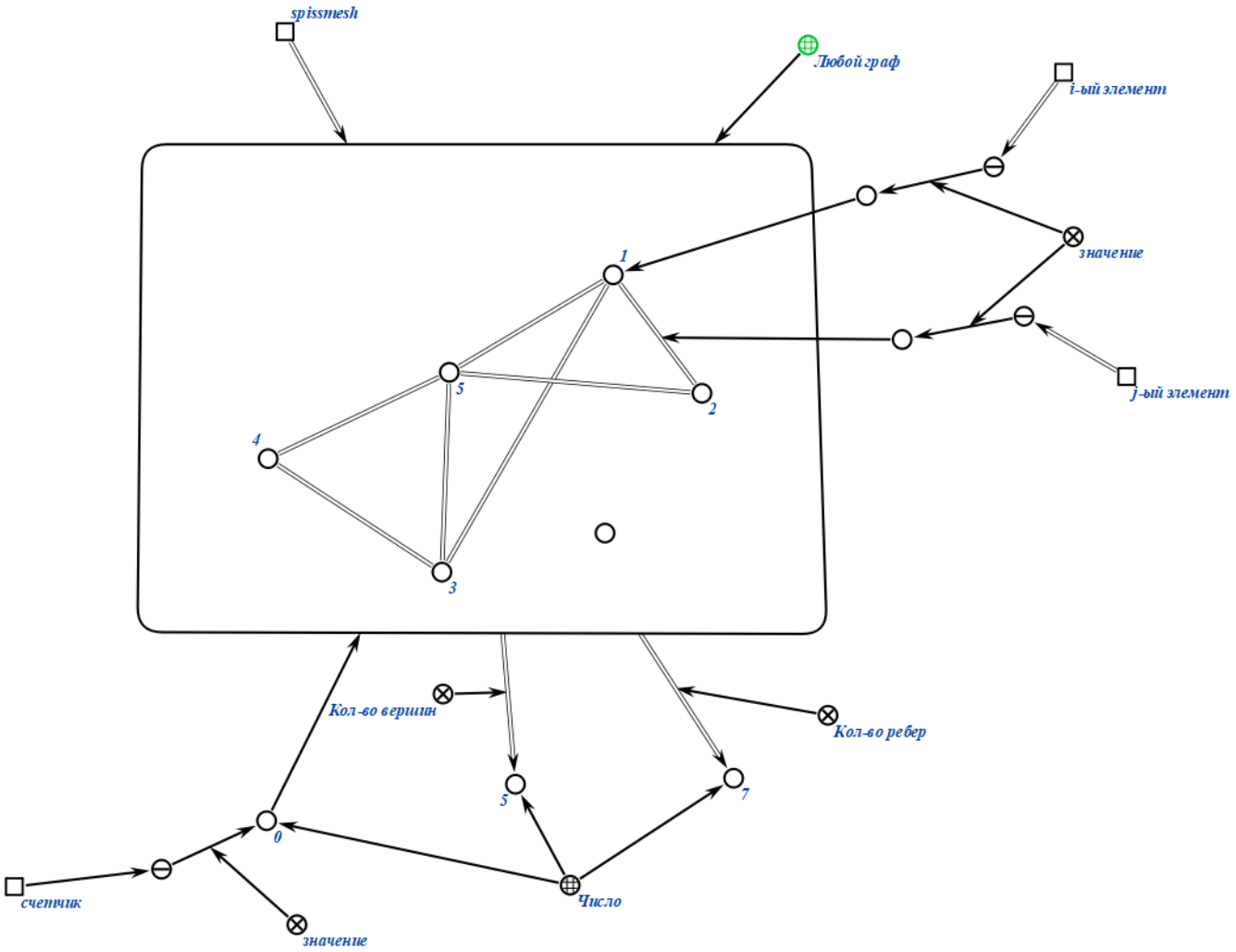

Рис. 16: Действие 2,3

4. Создаем объекты для перебора всех вершин и ребер, изначально он установлен на установлена на первую вершину и ребро

5. Счетчик пока остается на 0

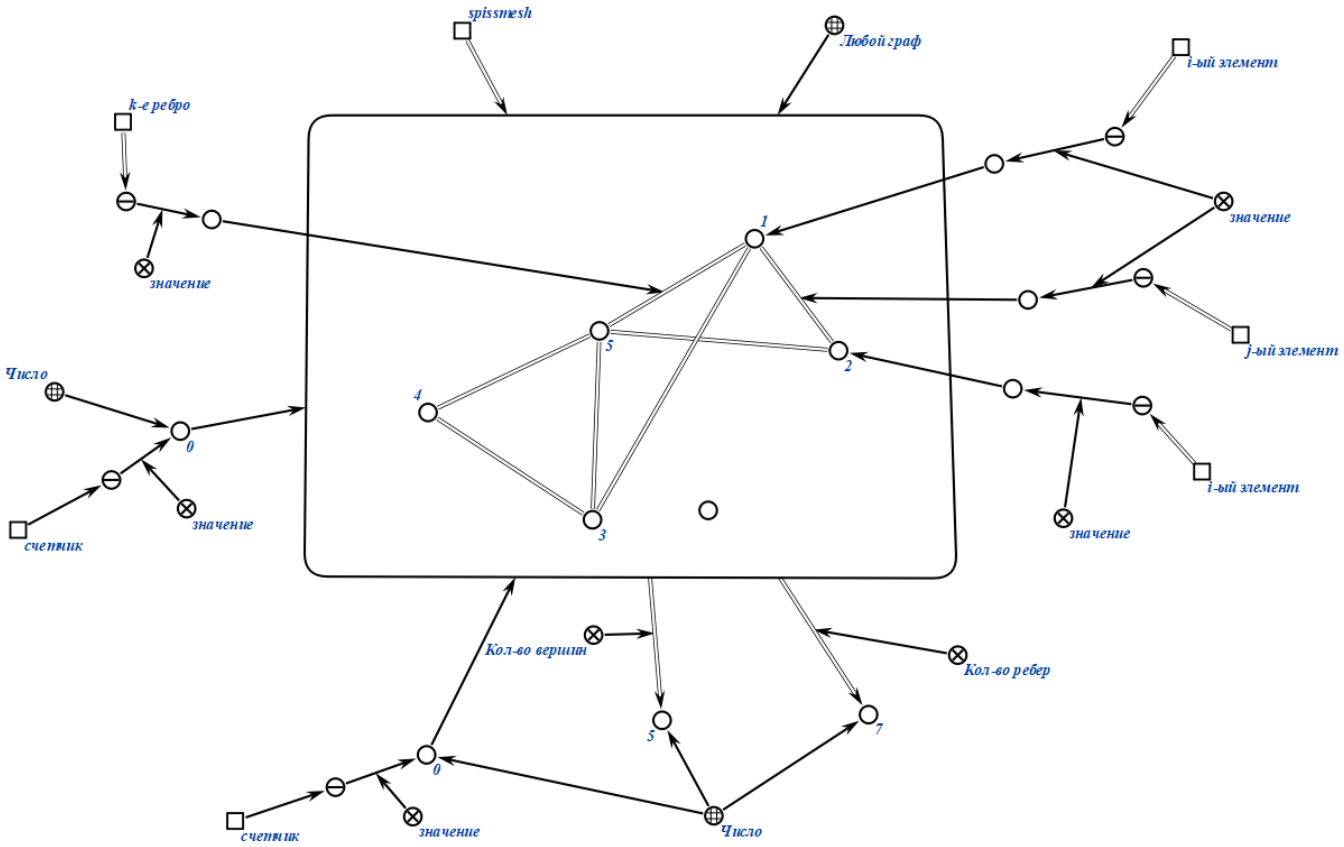

Рис. 17: Действие 6,7,8,9

6. Создаем объект для нахождения всех смежных ребер для кадого ребра;
7. Создаем счетчик смежных ребер для каждого ребра;
8. Пачинаем проверять первое ребро;
9. Проверяем до тех пор пока не найдем две вершины ;

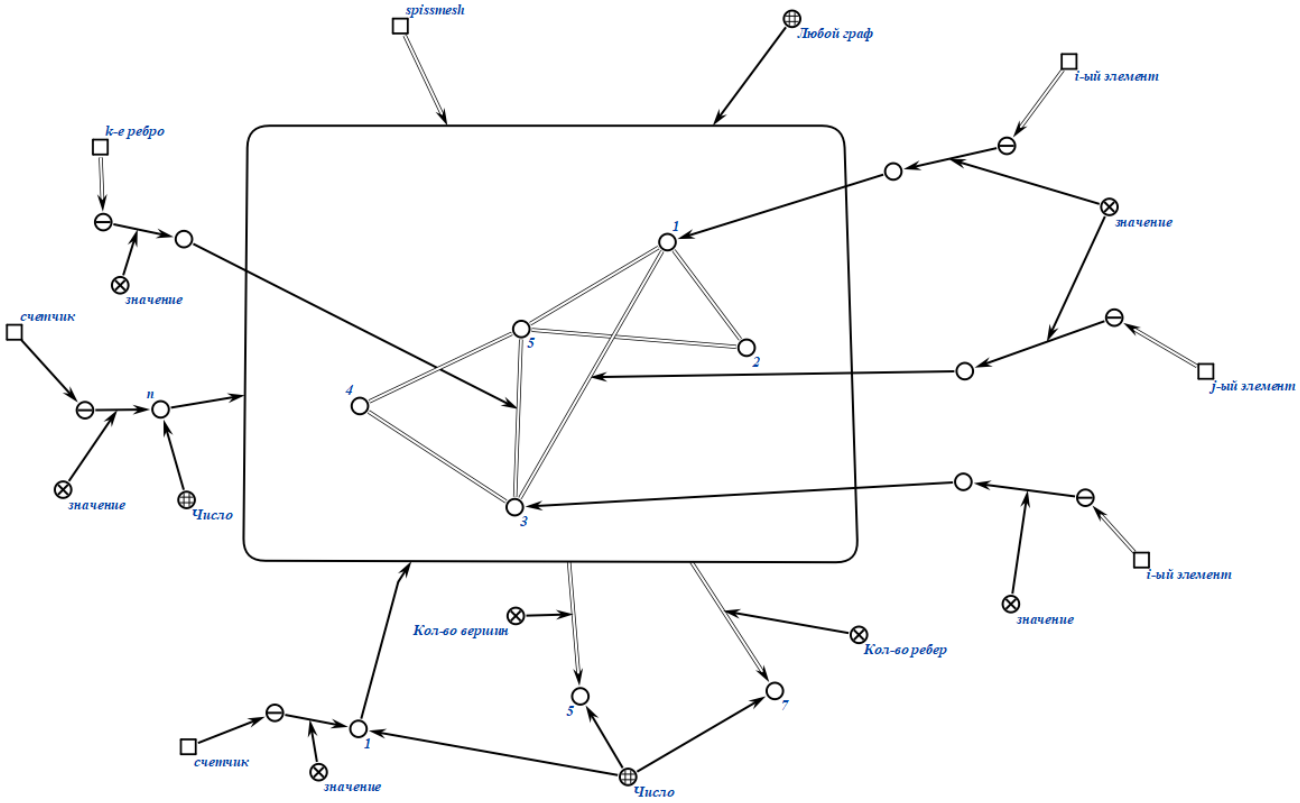

Рис. 18: Действие 10,11,12,13

10. Проверяем второе ребро и все оставшиеся с каждой вершиной;

11. Счетчик при это будет получать новое значение в зависимости от ребер каждый раз как будет проверено ребро;

12. После того, как мы проверили все ребра и нашли их количество, начинаем строить новый граф по следующим условиям:

12.1) Количество вершин нового графа должно быть равно количеству ребер изначального графа.

12.2) Граф неориетированный

13. Построим новый граф на примере этого.

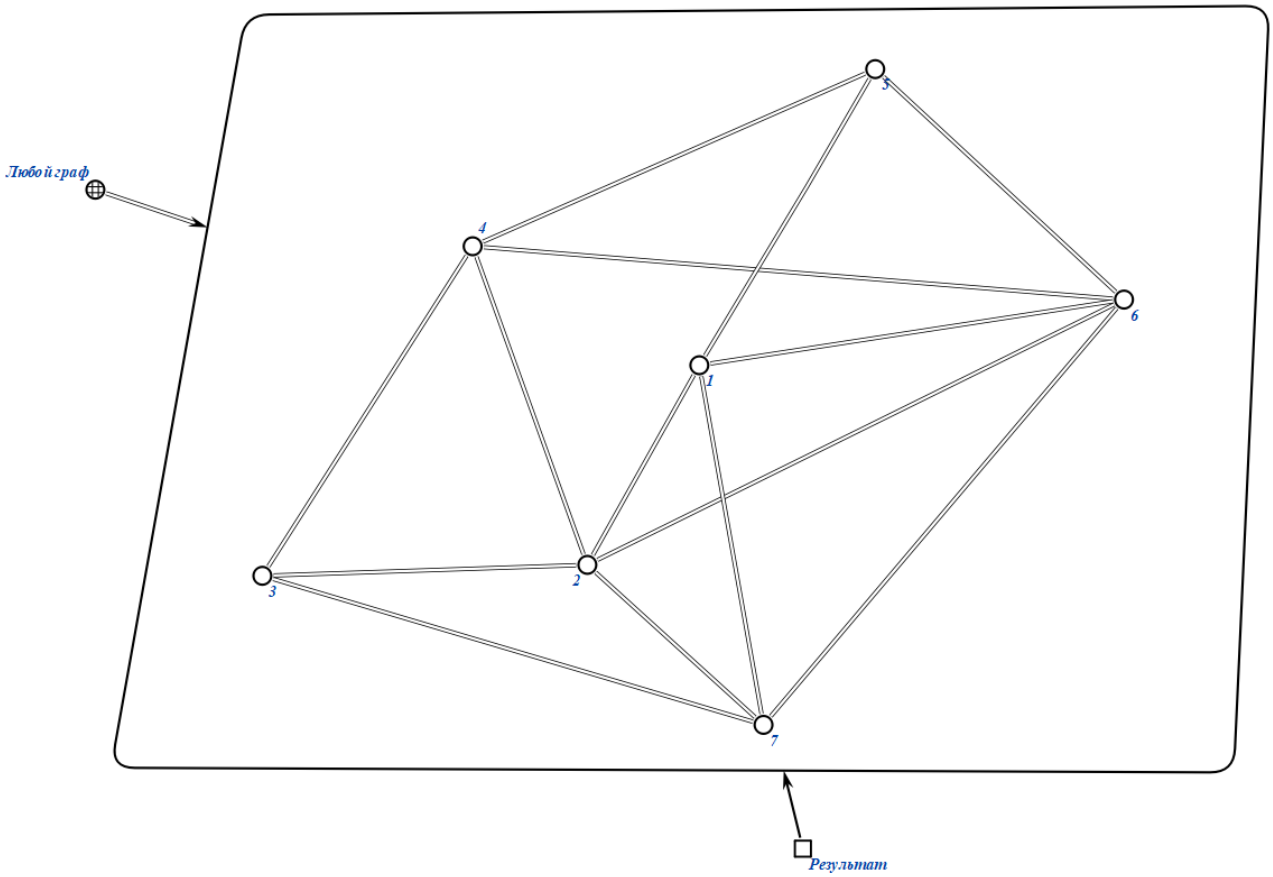

Рис. 19: Действие 14

14. Завершение алгоритма.

Заключение

В заключении у нас получилось формализовать поставленную задачу. Продемонстрировал графодинамику выполнения алгоритма.

Список литературы

[1] OSTIS GT [В Интернете] // База знаний по теории графов OSTIS GT. - 2011 г.. - http://ostisgraphstheo.sourceforge.net/index.php/

[2] Харарри Ф. Теория графов. Москва: ЕдиториалУРСС, 2003.

[3] Бхаргава, А. Грокаем алгоритмы : иллюстрированное пособие для программистов и любопытствующих / А. Бхаргава. — СПб : Питер, 2017. — 288 с.

[4] Оре, О. Теория графов / О. Оре. — Наука, 1980. — С. 336.

[5] Кормен, Д. Алгоритмы. Построение и анализ / Д. Кормен. — Вильямс, 2015. — С. 1328