

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 эйлеров цикл в неориентированном графе.**

Выполнил:

М. И. Курило

Студент группы  
321702

Проверил:

Н. В. Малиновская

Минск 2024

# Содержание

<b>1</b>	<b>Введение</b>	<b>2</b>
<b>2</b>	<b>Список понятий</b>	<b>2</b>
<b>3</b>	<b>Тестовые примеры</b>	<b>5</b>
3.1	Тест 1 . . . . .	5
3.2	Тест 2 . . . . .	6
3.3	Тест 3 . . . . .	7
3.4	Тест 4 . . . . .	8
<b>4</b>	<b>Пример работы алгоритма в семантической памяти</b>	<b>9</b>
4.1	Краткое описание: . . . . .	9
4.2	Демонстрация на тесте 5: . . . . .	9
<b>5</b>	<b>Заключение</b>	<b>15</b>

# 1 Введение

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

**Задача:** Найти эйлеров цикл в неориентированного графа.

## 2 Список понятий

1. **Неориентированный граф** (абсолютное понятие)-граф, в котором все ребра являются звеньями, то есть порядок двух концов ребра графа не существует

- (a) Вершина (относительное понятие, ролевое отношение);
- (b) Связка (относительное понятие, ролевое отношение).

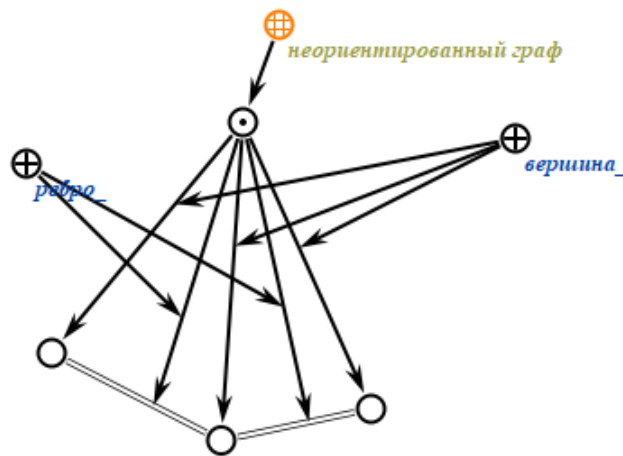


Рис. 1: Абсолютное понятие неориентированного графа

2. *Путь в графе* — последовательность вершин, в которой каждая вершина соединена со следующей ребром.

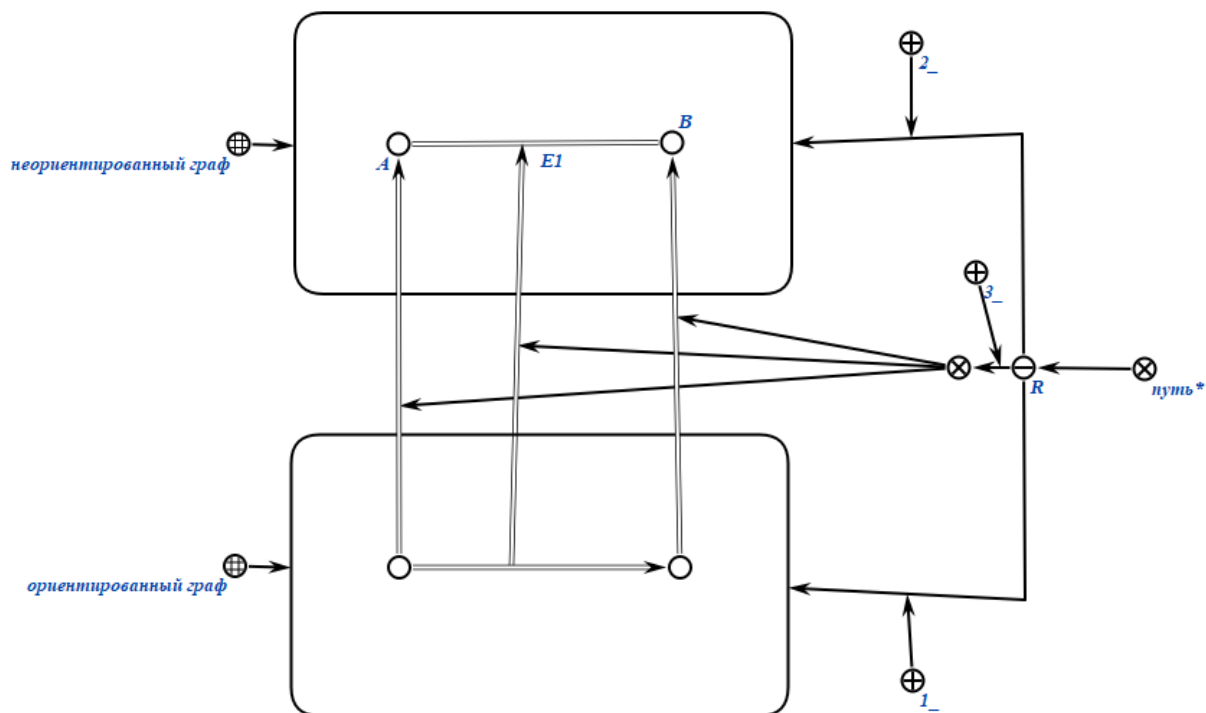


Рис. 2: Абсолютное понятие диаметра

3. **Эйлеровым циклом** называется замкнутый путь, проходящий через каждое ребро графа ровно по одному разу.

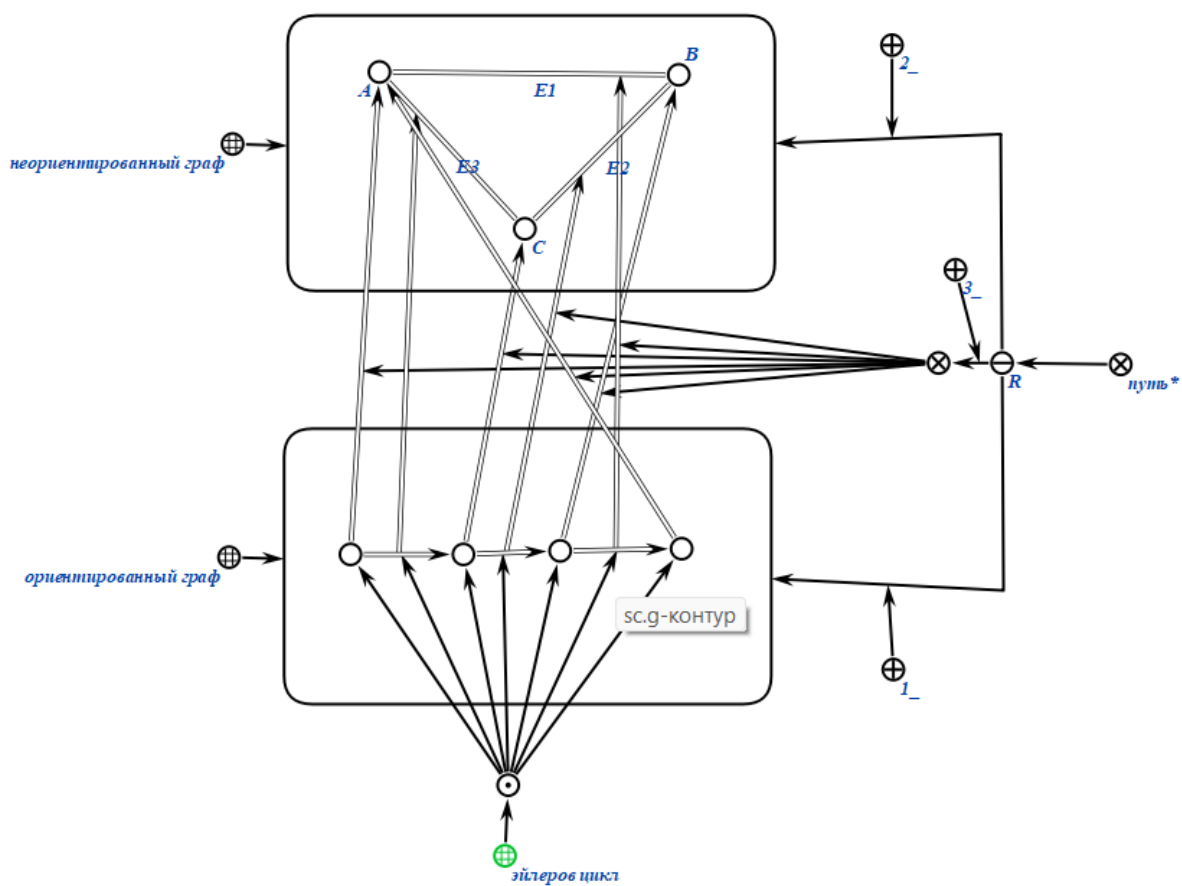


Рис. 3: Абсолютное понятие эксцентриситета

### 3 Тестовые примеры

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

#### 3.1 Тест 1

**Вход:** Необходимо найти эйлеров цикл в графе

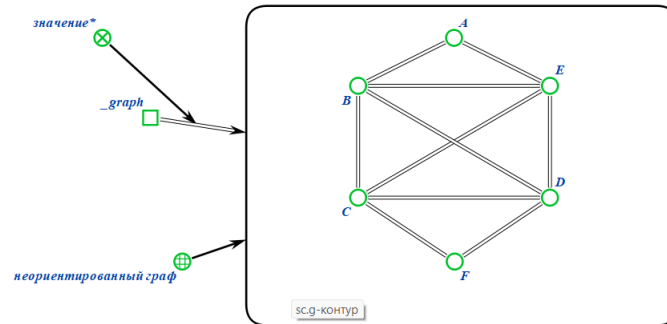


Рис. 4: Вход теста 1

**Выход:** Будет найден порядок прохождения ребер и вершин.

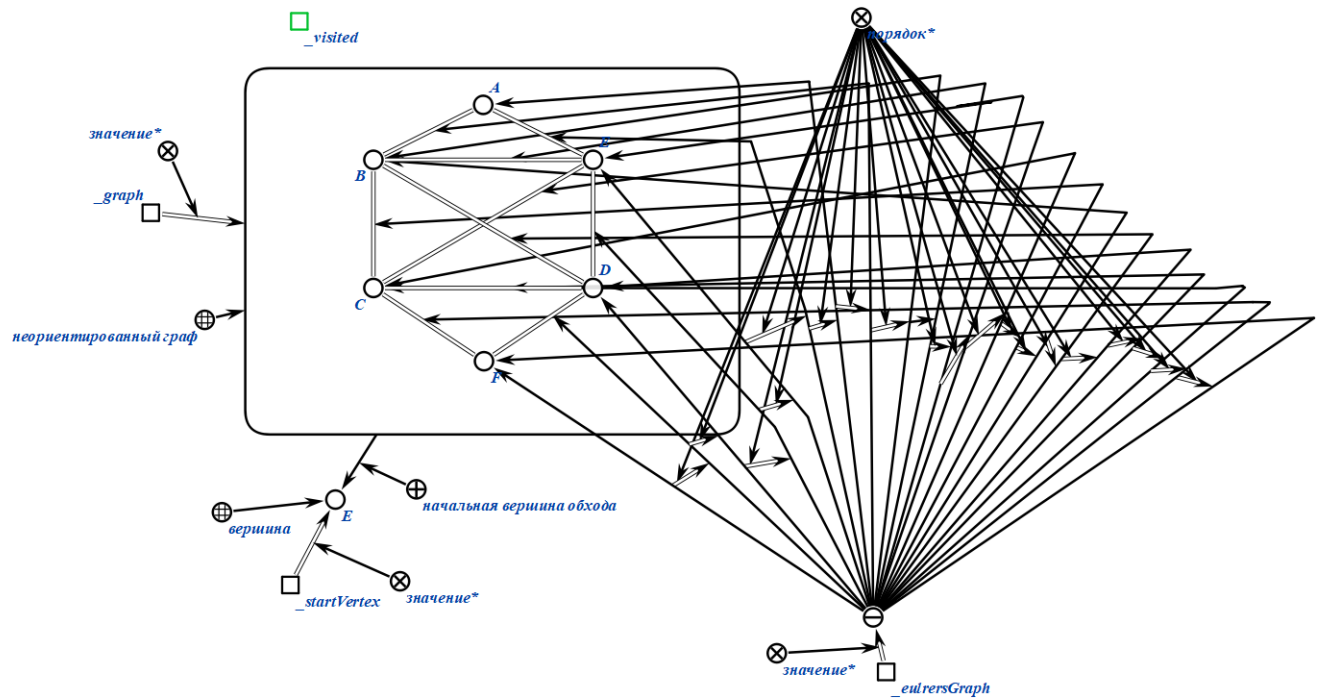


Рис. 5: Выход теста 1

### 3.2 Тест 2

**Вход:** Необходимо найти эйлеров цикл в графе.

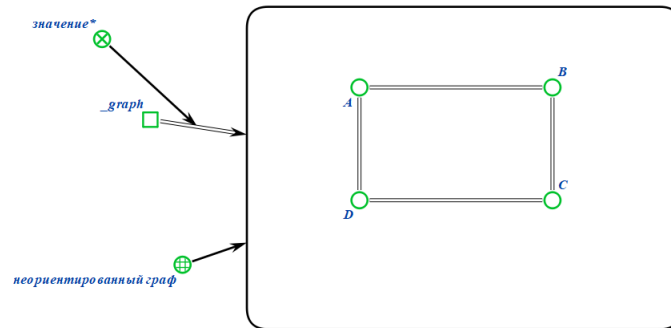


Рис. 6: Выход теста 2

**Выход:** Будет найден порядок прохождения ребер и вершин.

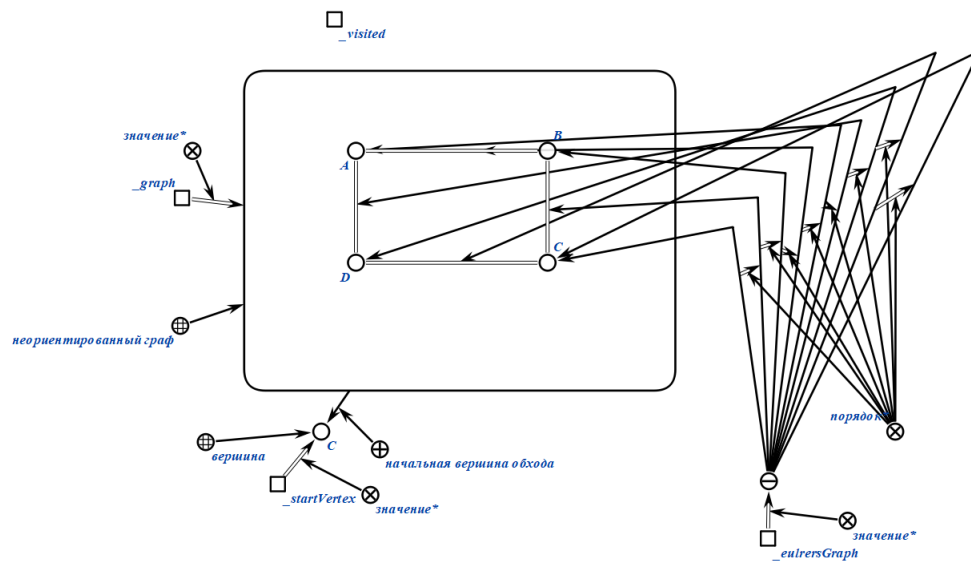


Рис. 7: Вход теста 2

### 3.3 Тест 3

**Вход:** Необходимо найти эйлеров цикл в графе.

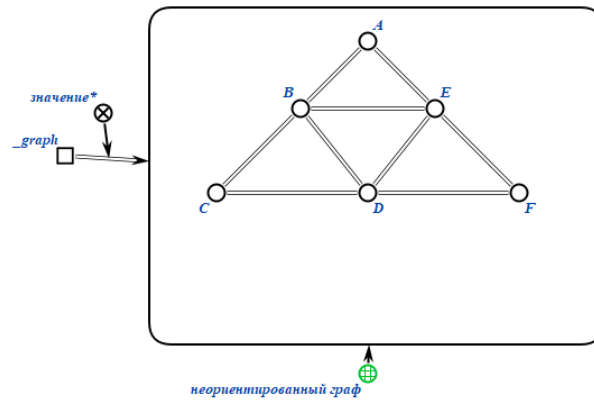


Рис. 8: Вход теста 3

**Выход:** Будет найден порядок прохождения ребер и вершин.

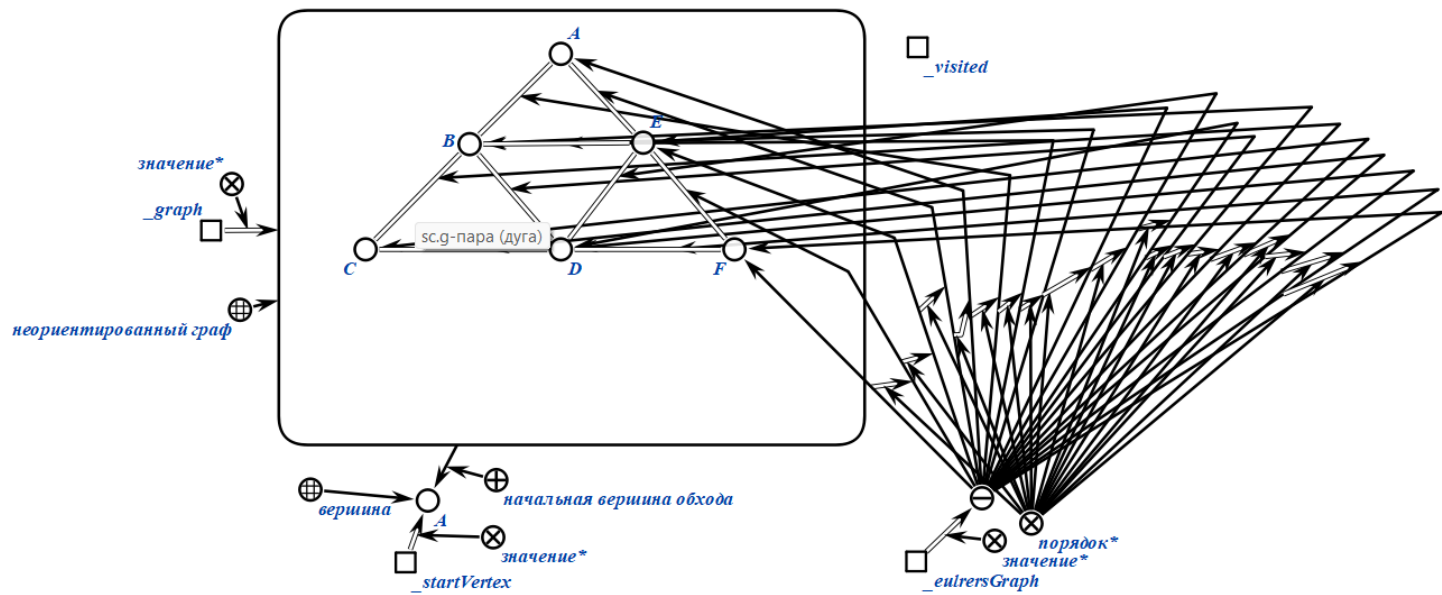


Рис. 9: Вход теста 3



### 3.4 Тест 4

**Вход:** Необходимо найти эйлеровов цикл в графе.

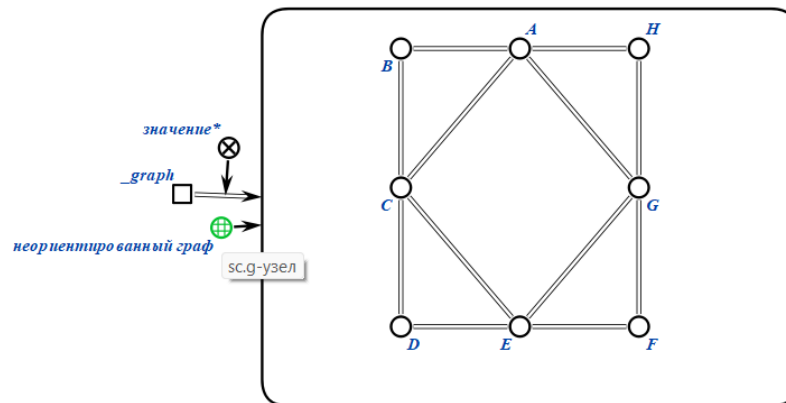


Рис. 10: Вход теста 4

**Выход:** Будет найден порядок прохождения ребер и вершин.

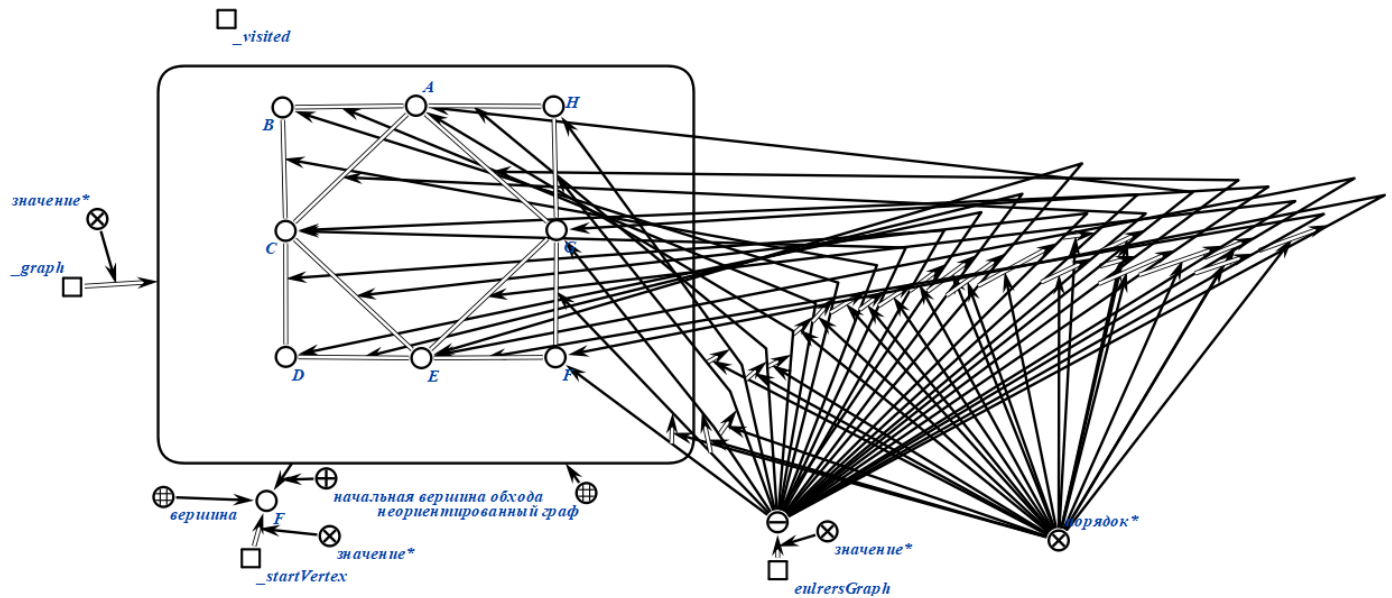


Рис. 11: Выход теста 4

## 4 Пример работы алгоритма в семантической памяти

### 4.1 Краткое описание:

1. Пользователь задает неориентированный граф.
2. Пользователь выбирает вершину для начала обхода.
3. При помощи DFS и двух списков (`visited`, `eulersGraph`) находится эйлеров цикл, если он существует. Обход начинается с заданной вершины. Направление обхода случайное, после перехода от одной вершины к другой, ребро между ними "стирается" а вершина, к которой был выполнен переход, записывается в список `visited`. Если у вершины нет ребер, то она переносится в список `eulersGraph`.
4. Результат представляет собой список вершин: `eulersGraph`.

### 4.2 Демонстрация на тесте 5:

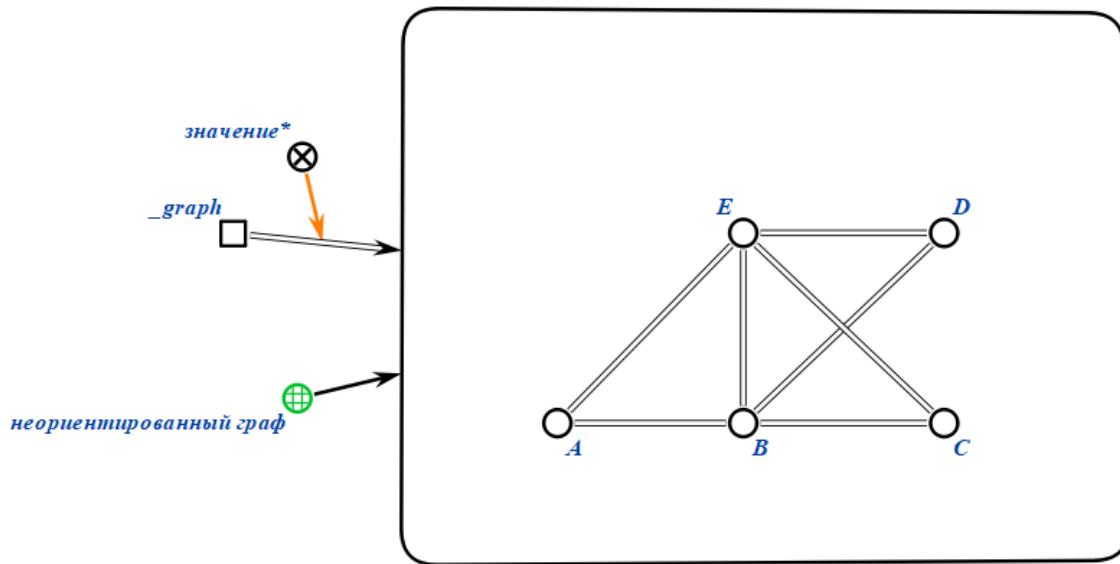


Рис. 12: Вход теста 5

1. Пользователь выбирает вершину для начала обхода (предположим, A);

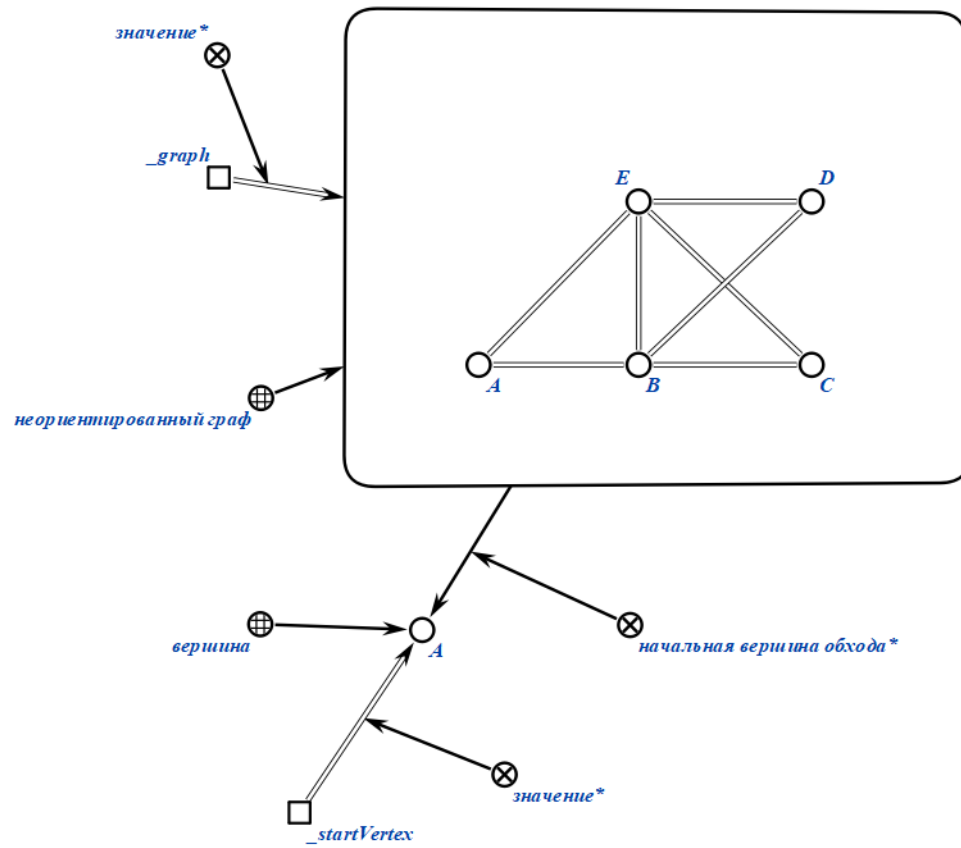


Рис. 13: Действие 1

2. Начинаем обход с вершины A, заносим ее в список visited.

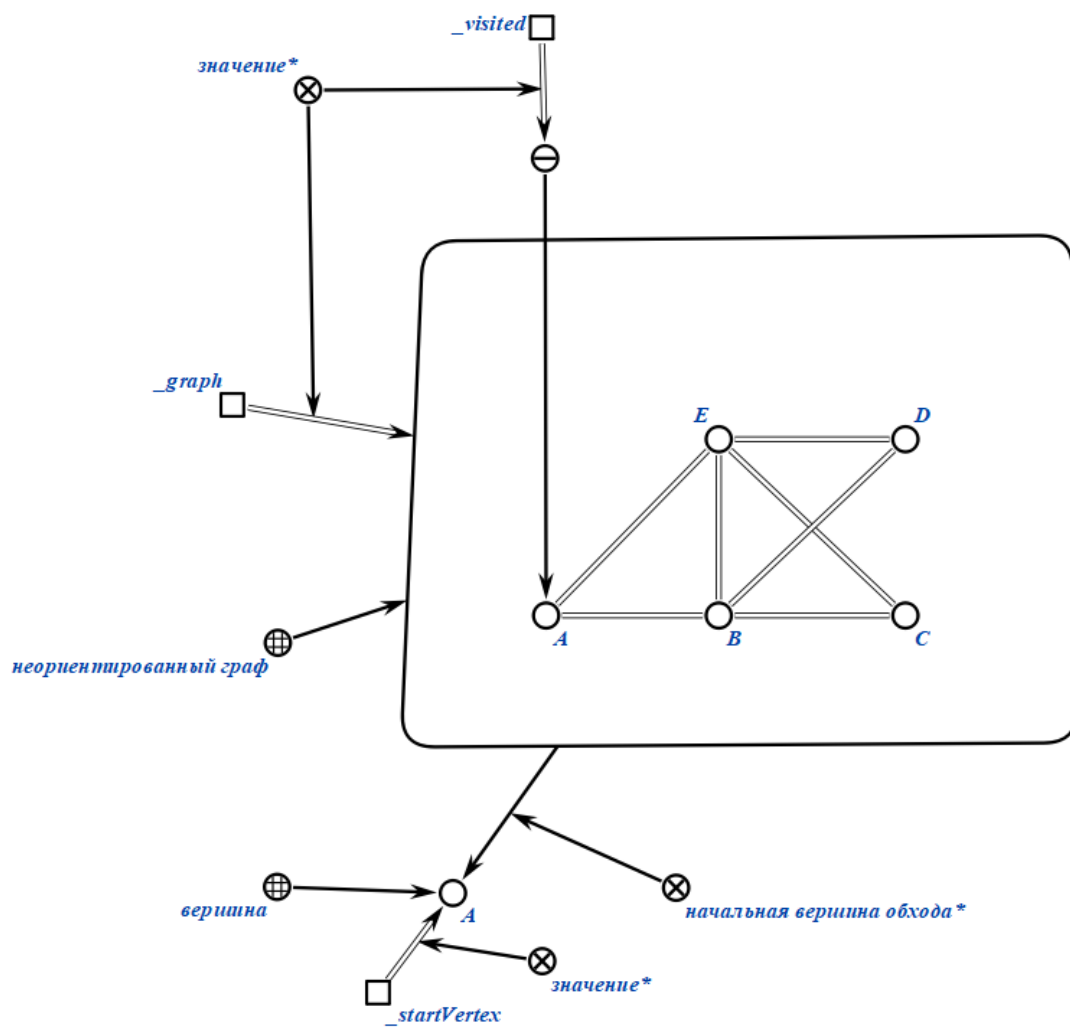


Рис. 14: Действие 2

3. Идем к следующей вершине: E, убирая ребро AE и записывая E в список visited.

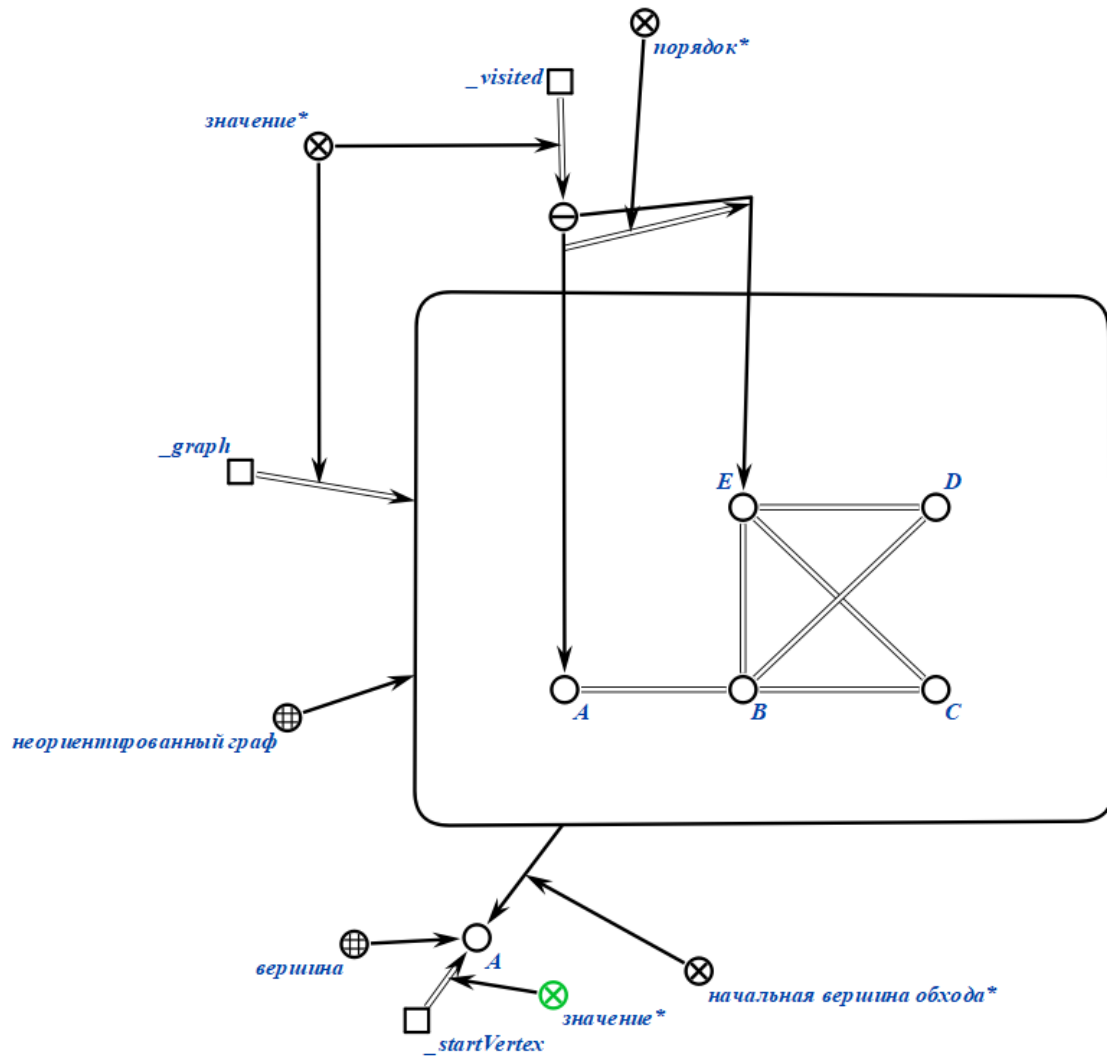


Рис. 15: Действие 3

4. Повторяем действие 3, пока не доходим до вершины, из которой "выхода" нет: A.

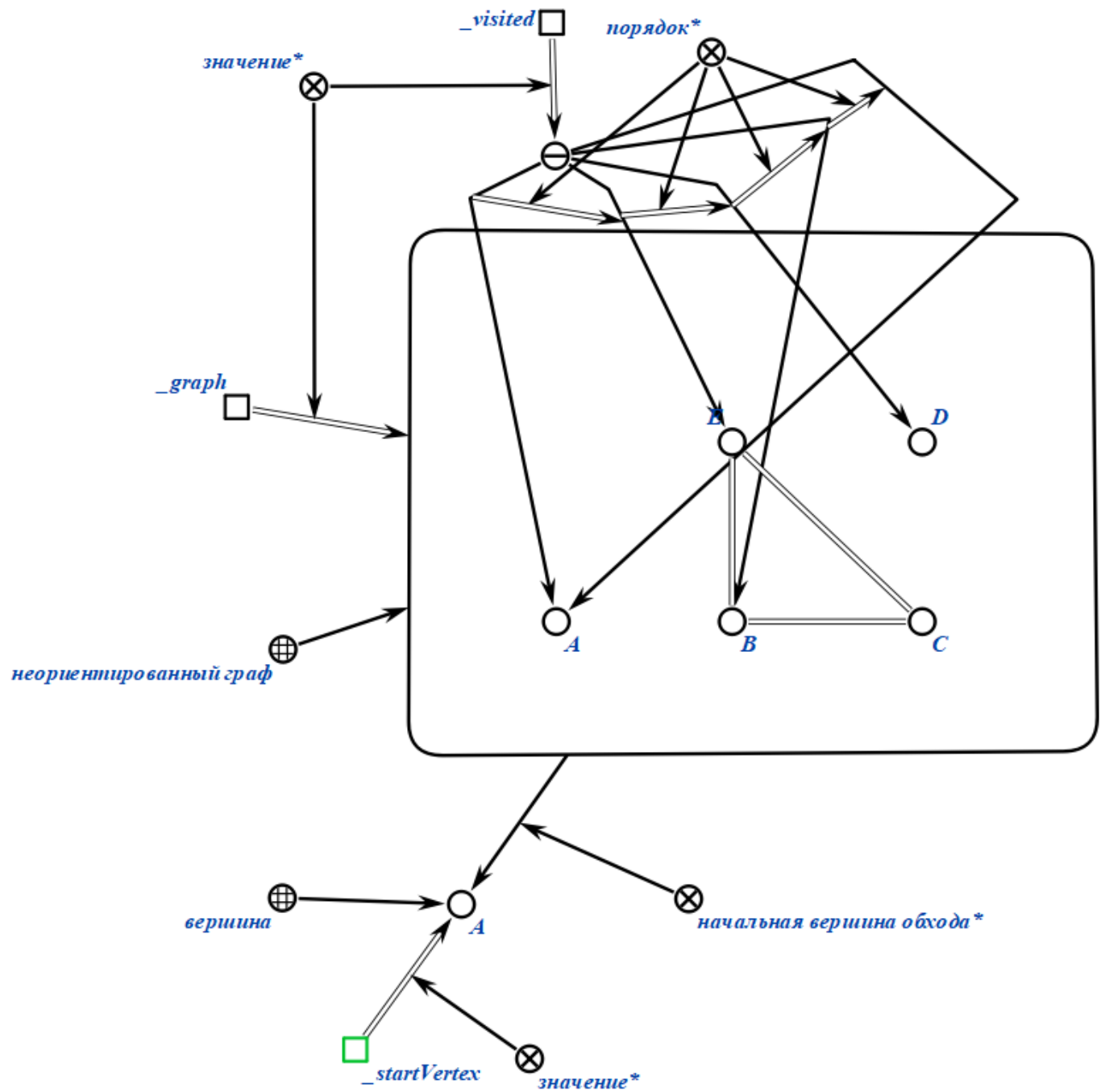


Рис. 16: Действие 4

5. Следуя нашему алгоритму, переносим вершину A в список eulersGraph и возвращаемся к вершине, из которой мы пришли. Продолжаем алгоритм.

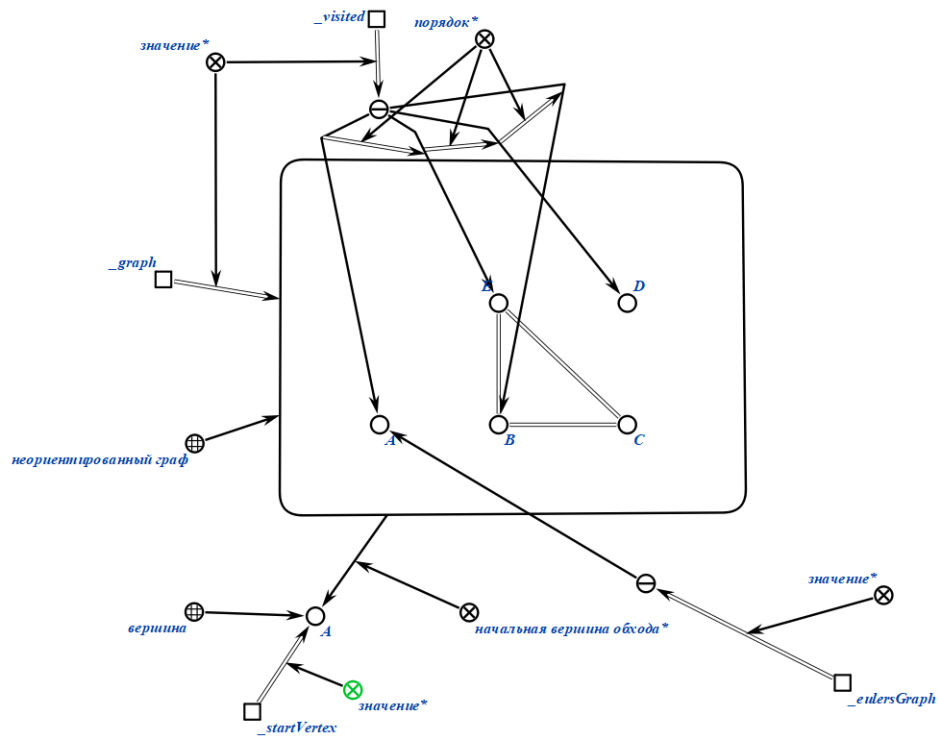


Рис. 17: Действие 5

6. Формируем граф, который является ответом, соединяя вершины в том порядке, как они записаны.

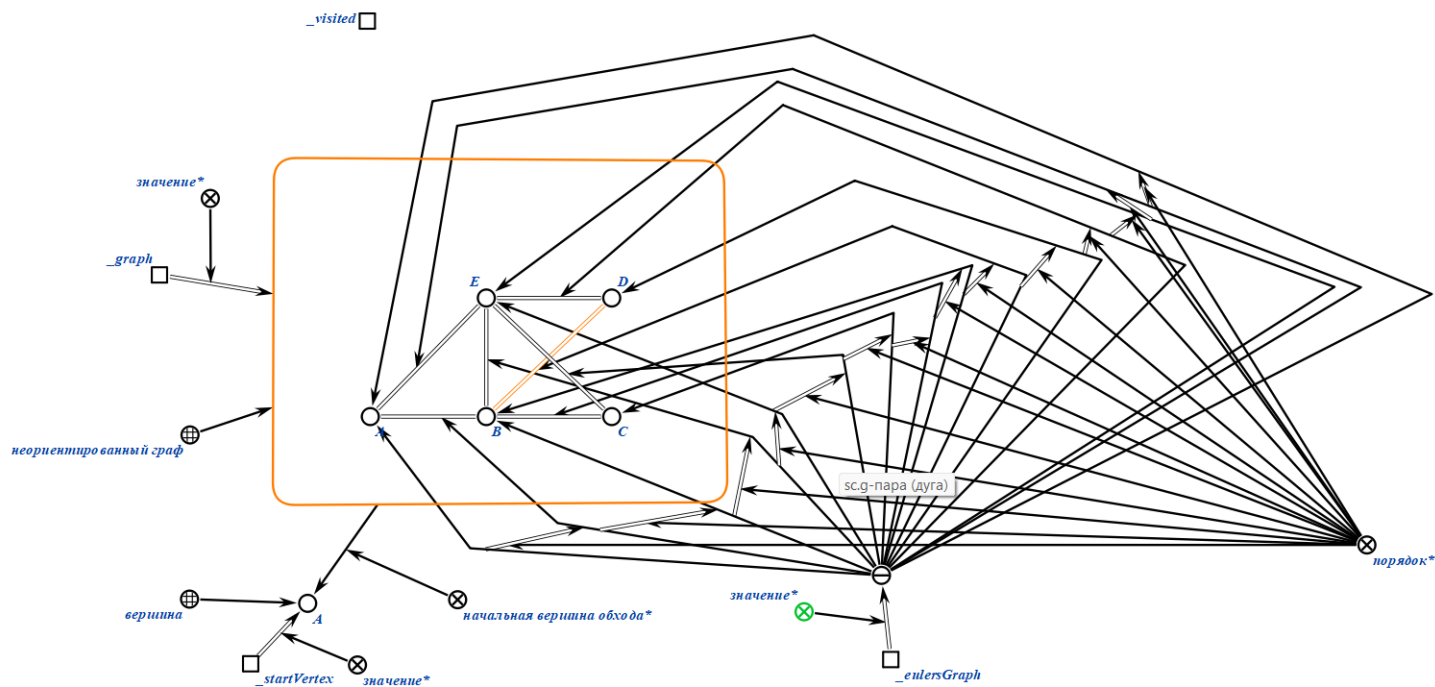


Рис. 18: Действие 6

## 5 Заключение

В заключении у нас получилось формализовать поставленную задачу. Мы нашли нужные нам циклы. Реализовали алгоритм их поиска, который работает на любом неориентированном графе.