

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДРАСТВЕННЫЙ 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 поиска точек сочленения в неориентированном графе

Выполнил:

В.Д. Хорошавин

Студент группы
021704

Проверил:

В.С. Витязь

Минск 2021

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

Задача: Поиск точек сочленения в неориентированном графе

1 СПИСОК ПОНЯТИЙ

1. Графовая структура (абсолютное понятие) - это такая одноуровневая реляционная структура, объекты которой могут играть роль либо вершины, либо связки:

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

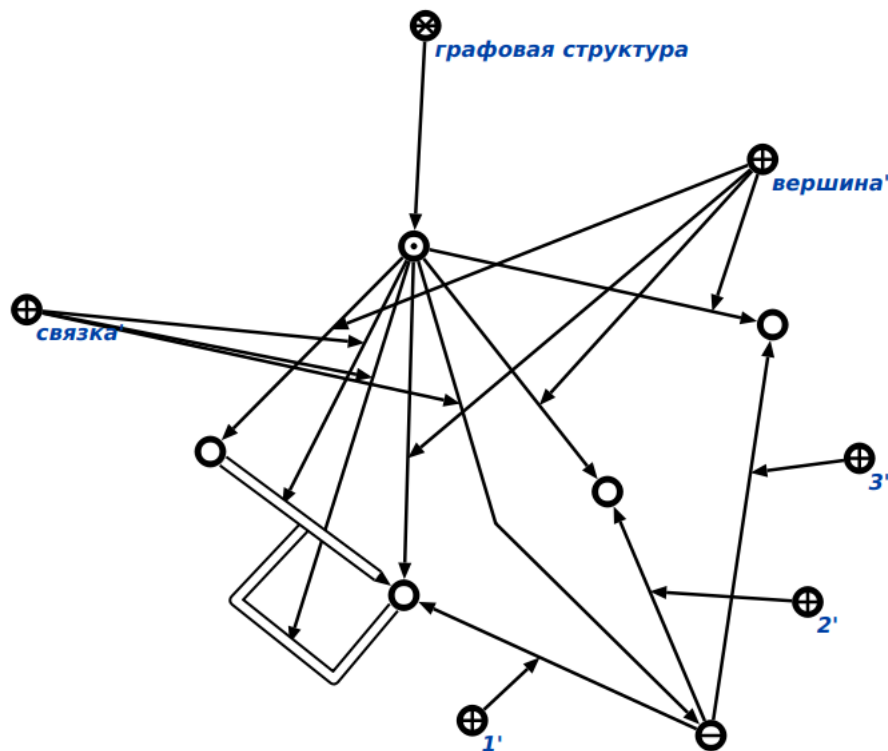


Рис. 1: Графовая структура

2. Графовая структура с ориентированными связками (абсолютное понятие)

- (a) Ориентированная связка (относительное понятие, ролевое отношение) – связка, которая задается ориентированным множеством.

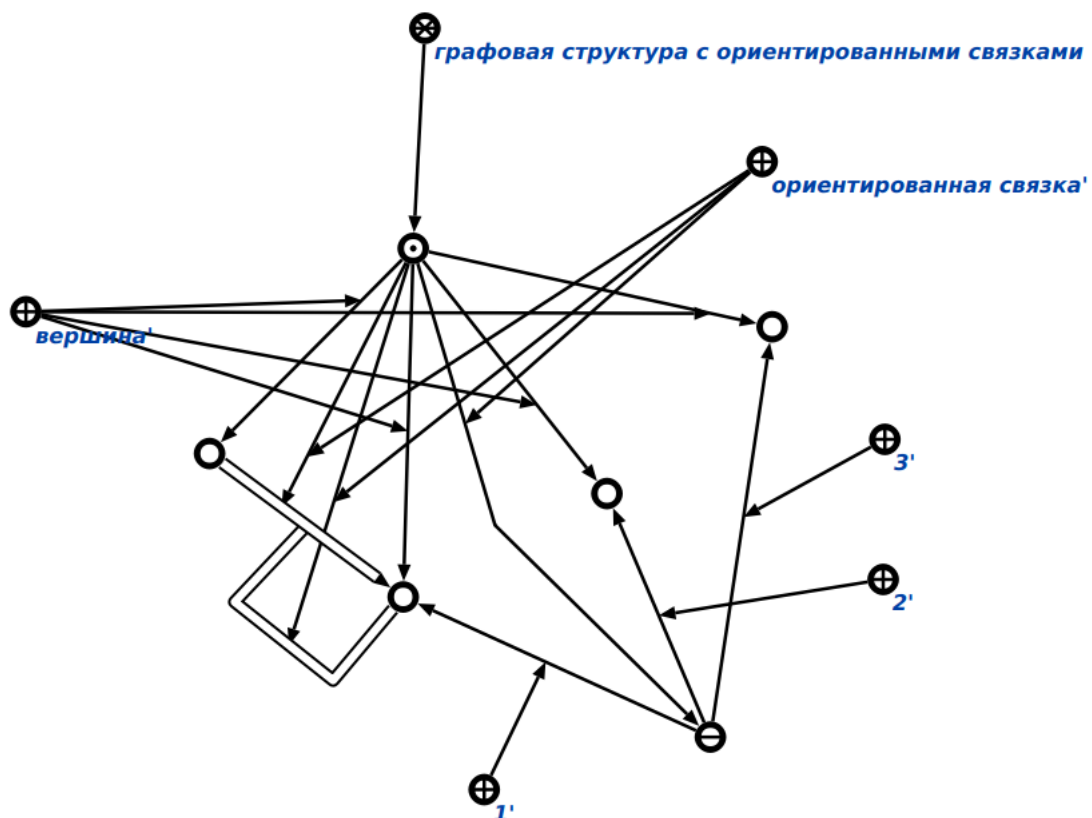


Рис. 2: Графовая структура с ориентированными связками

3. Графовая структура с неориентированными связками (абсолютное понятие)

(а) Неориентированная связка (относительное понятие, ролевое отношение) – связка, которая задается неориентированным множеством.

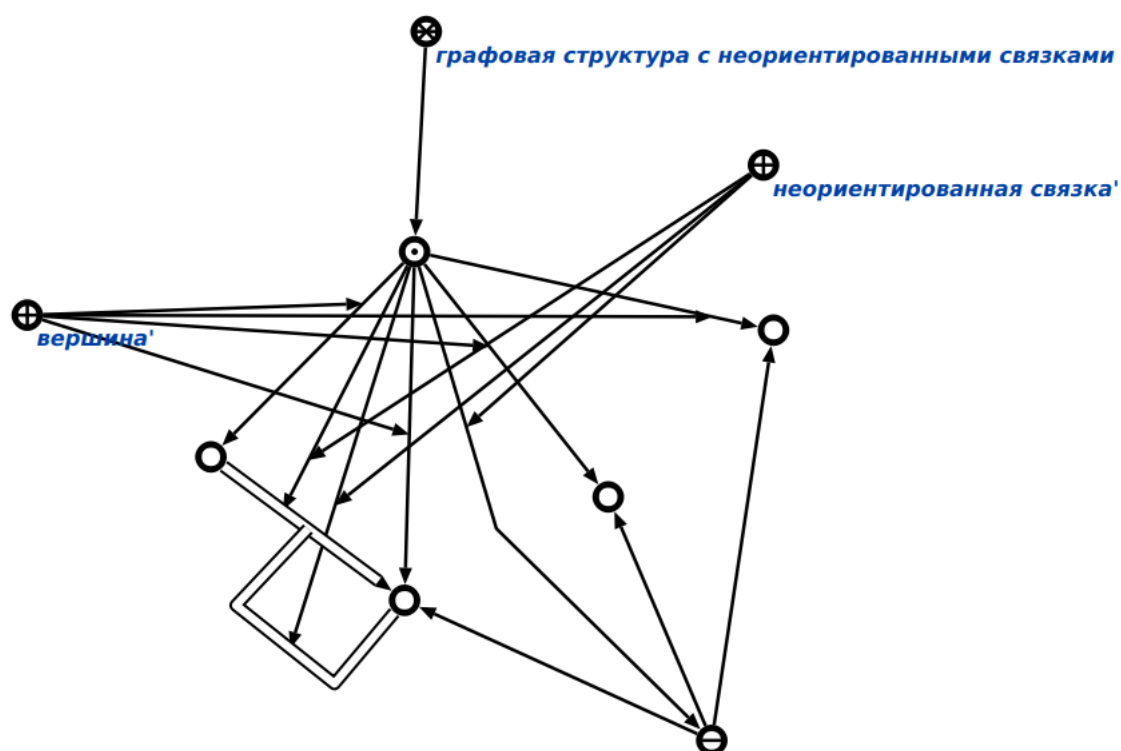


Рис. 3: Графовая структура с неориентированными связками

4. Гиперграф (абсолютное понятие) – это такая графовая структура, в которой связки могут связывать только вершины:
 - (а) Гиперсвязка (относительное понятие, ролевое отношение);
 - (b) Гипердуга (относительное понятие, ролевое отношение) – ориентированная гиперсвязка;
 - (с) Гиперребро (относительное понятие, ролевое отношение) – неориентированная гиперсвязка.

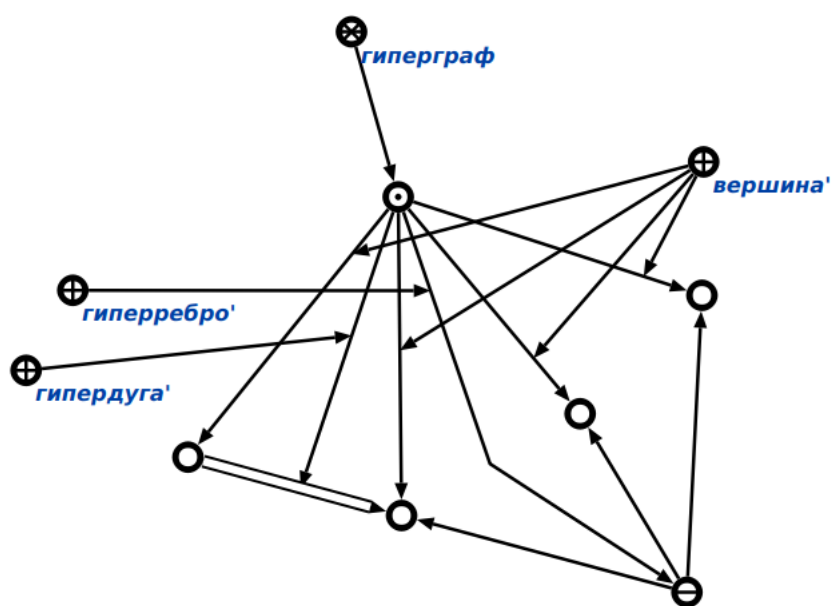


Рис. 4: Гиперграф

5. Псевдограф (абсолютное понятие) – это такой гиперграф, в котором все связки должны быть бинарными:

- (а) Бинарная связка (относительное понятие, ролевое отношение) – гиперсвязка арности 2;
- (b) Ребро (относительное понятие, ролевое отношение) – неориентированная гиперсвязка;
- (с) Дуга (относительное понятие, ролевое отношение) – ориентированная гиперсвязка;
- (d) Петля (относительное понятие, ролевое отношение) – бинарная связка, у которой первый и второй компоненты совпадают.

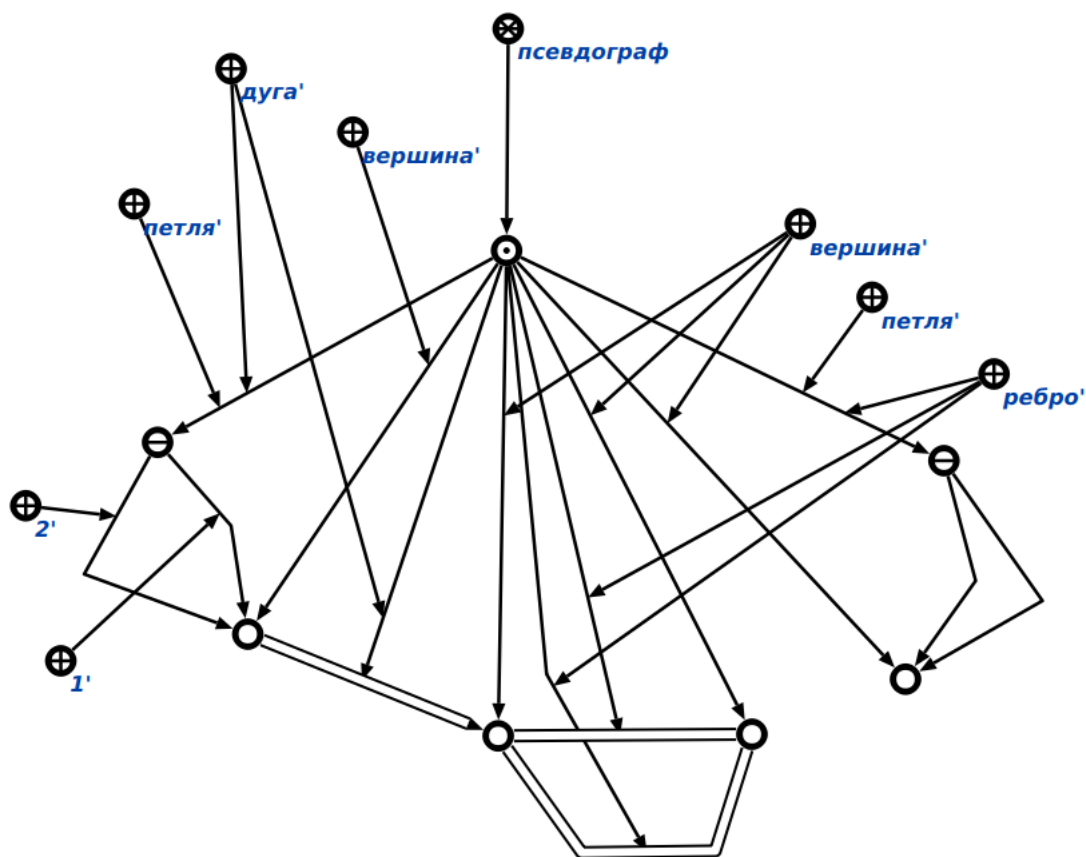


Рис. 5: Псевдограф

6. Мультиграф (абсолютное понятие) – это такой псевдограф, в котором не может быть петель:

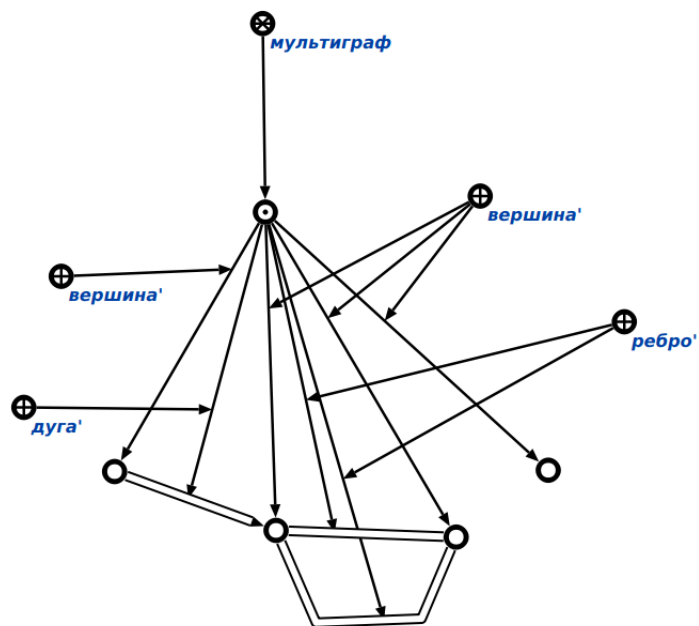


Рис. 6: Мультиграф

7. Граф (абсолютное понятие) – это такой мультиграф, в котором не может быть кратных связок, т.е. связок у которых первый и второй компоненты совпадают:

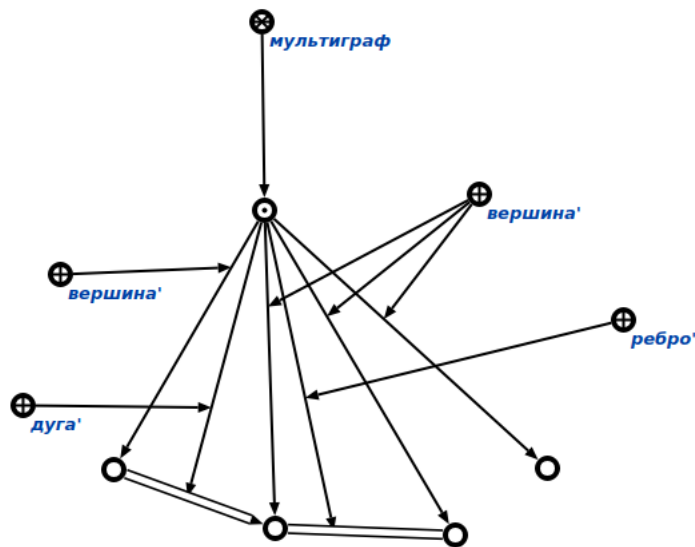


Рис. 7: Граф

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

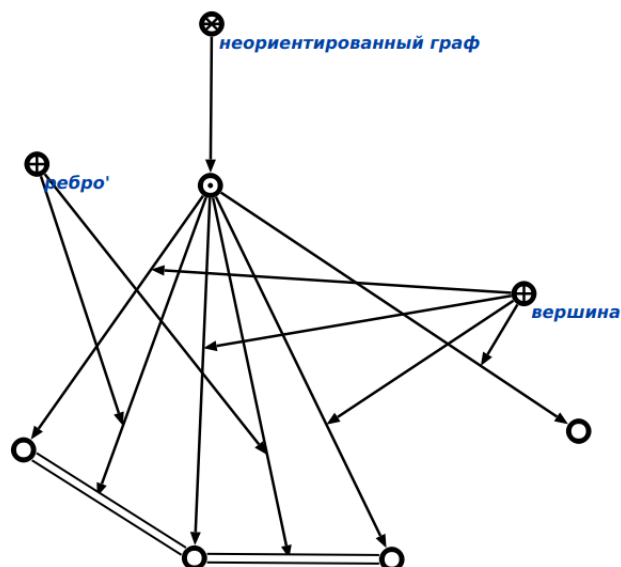


Рис. 8: Неориентированный граф

9. Связанный граф (абсолютное понятие) - граф, в котором между любой парой вершин есть путь:

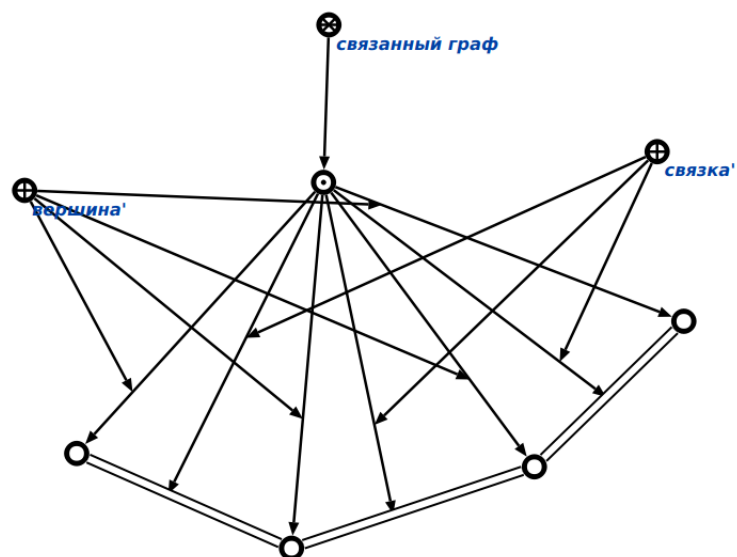


Рис. 9: Связанный граф

10. Точка сочленения (абсолютное понятие) - вершина при удалении которой из графа, он становится несвязанным:

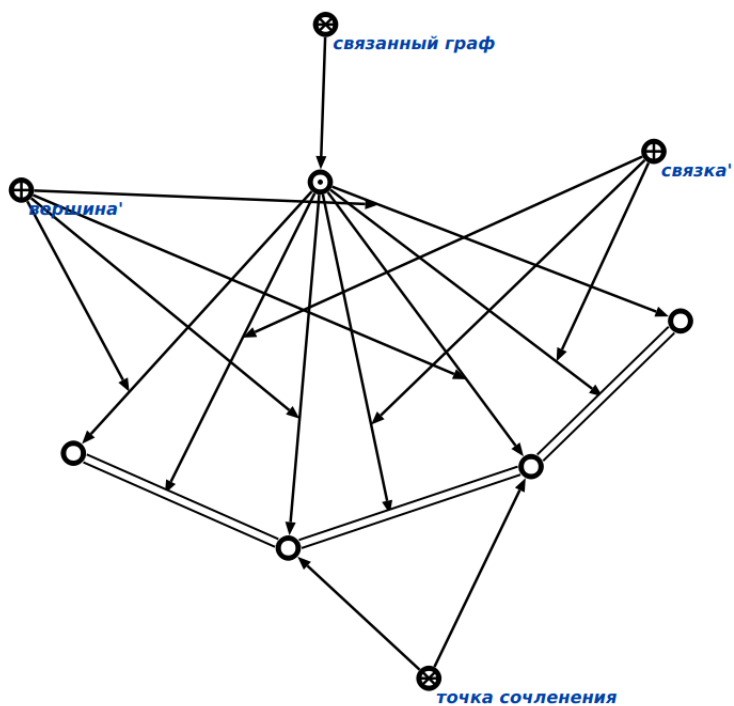


Рис. 10: Точка сочленения

2 ТЕСТОВЫЕ ПРИМЕРЫ

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

Во всех тестах решается задача поиска всех точек сочленения.

2.1 Тест 1

Вход:

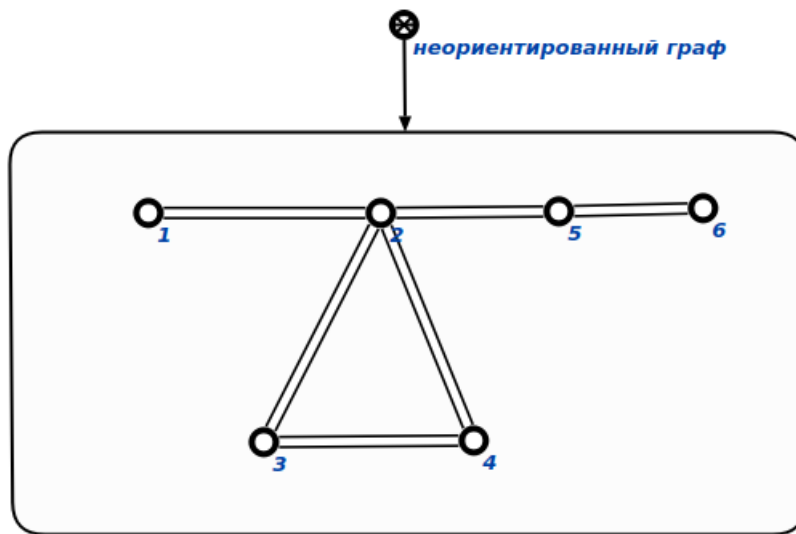


Рис. 11: Вход теста 1

Выход:

Будут найдены вершины 2 и 5:

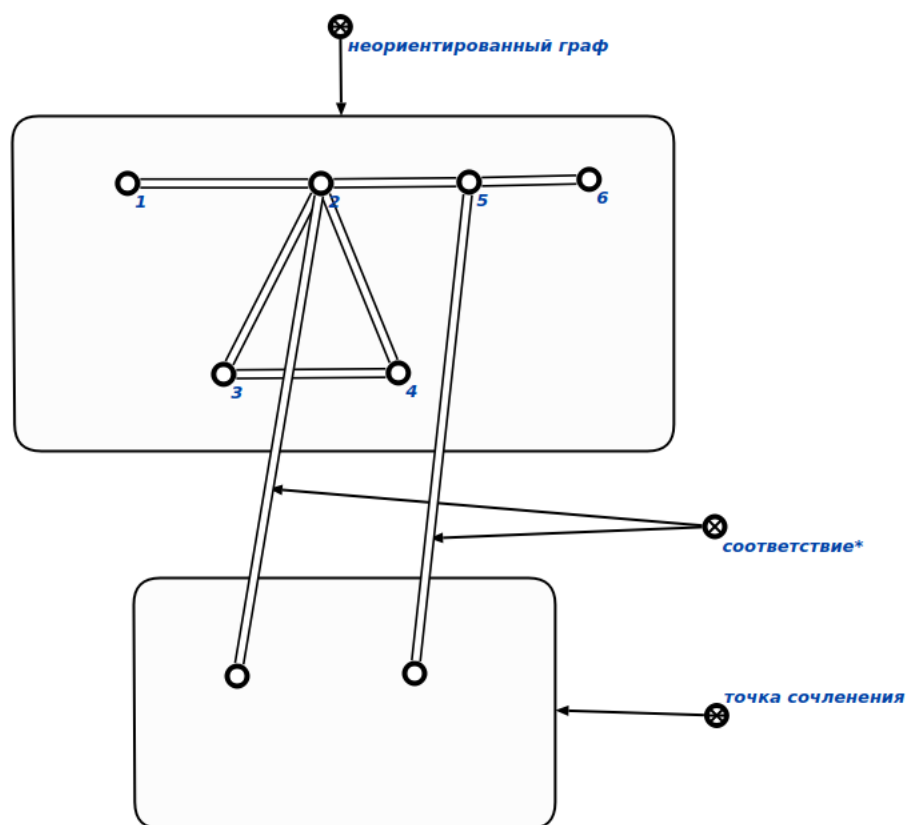


Рис. 12: Выход теста 1

2.2 Тест 2

Вход:

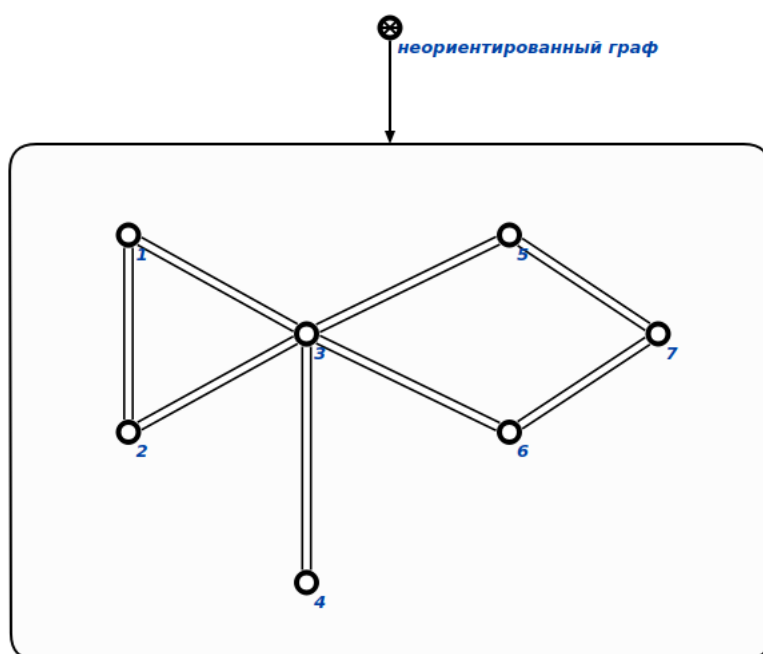


Рис. 13: Вход теста 2

Выход:

Будет найдена вершина 3:

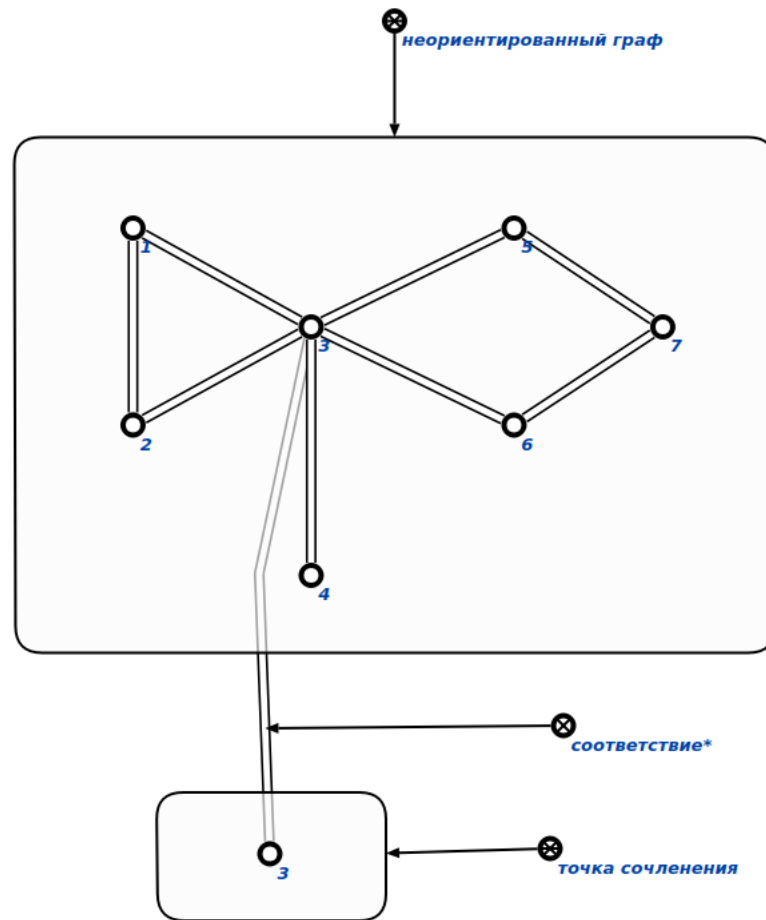


Рис. 14: Выход теста 2

2.3 Тест 3

Вход:

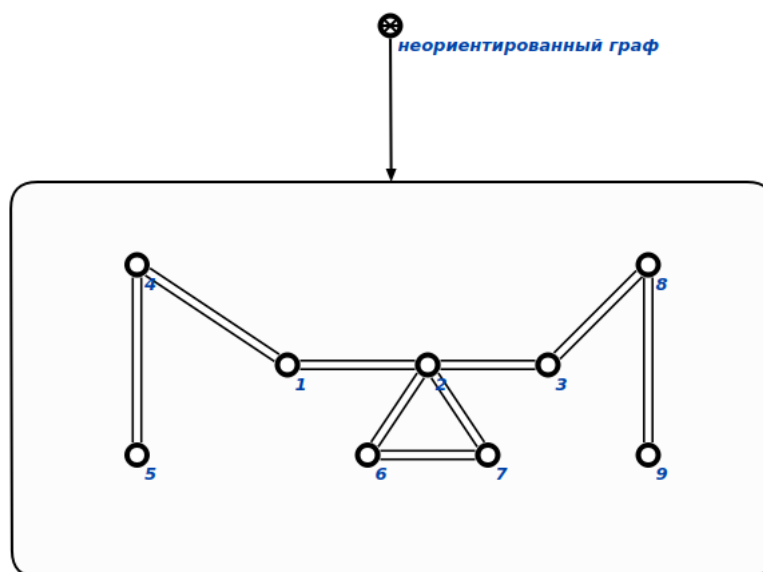


Рис. 15: Вход теста 3

Выход:

Будут найдены вершины 1, 2, 3, 4, 8:

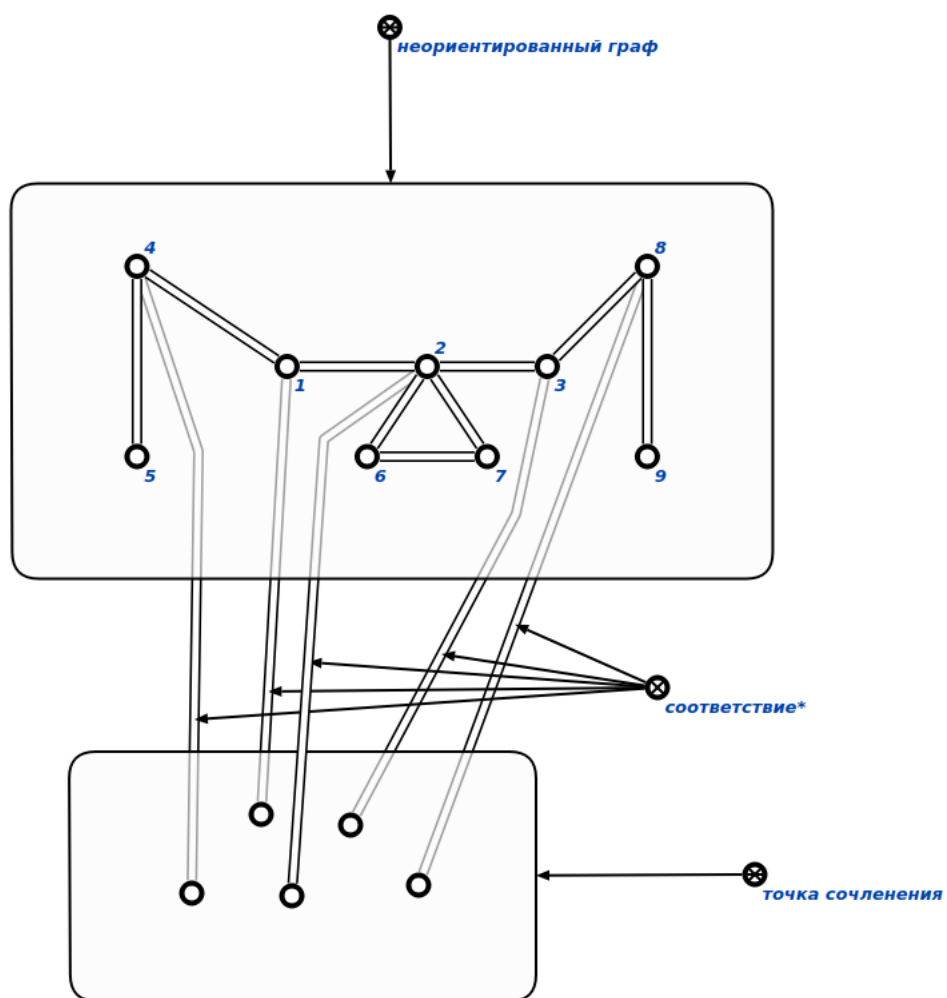


Рис. 16: Выход теста 3

2.4 Тест 4

Вход:

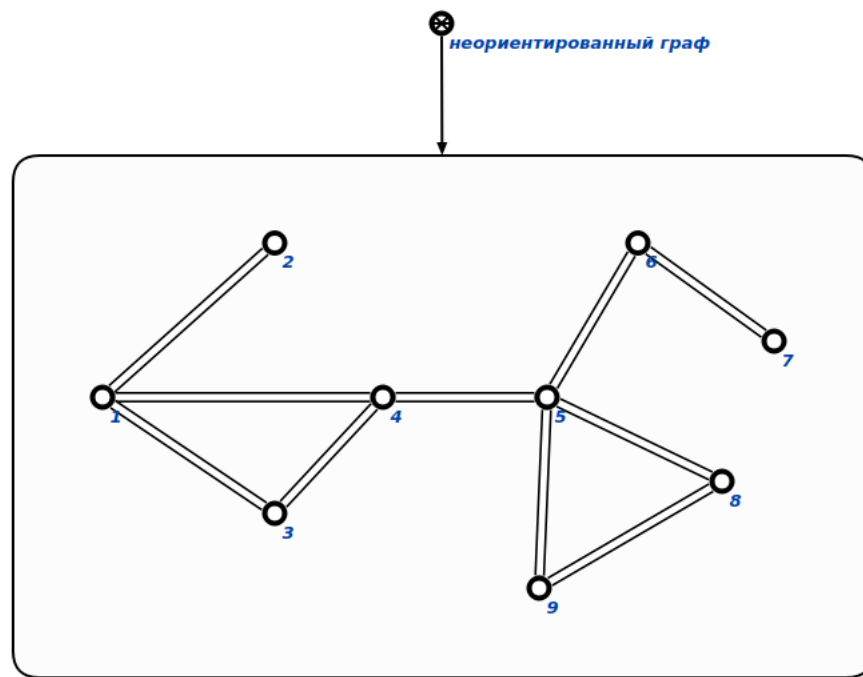


Рис. 17: Вход теста 4

Выход:

Будут найдены вершины 1, 4, 5, 6:

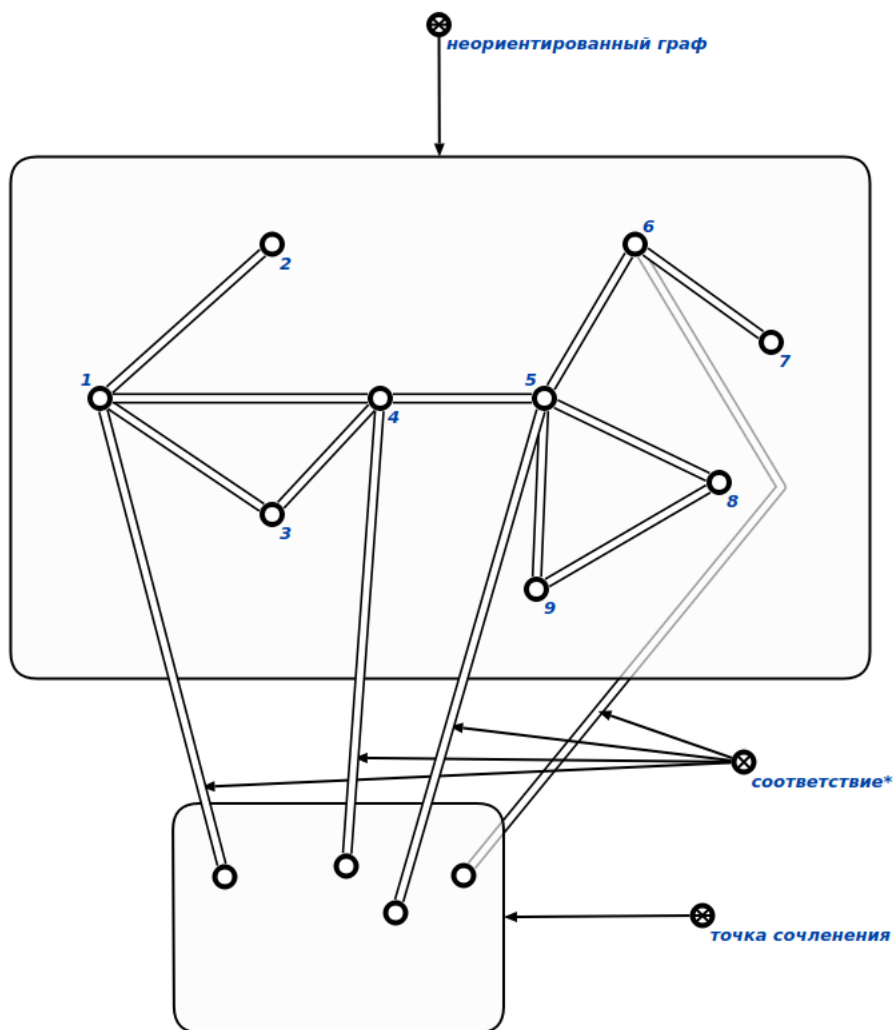


Рис. 18: Выход теста 4

2.5 Тест 5

Вход:

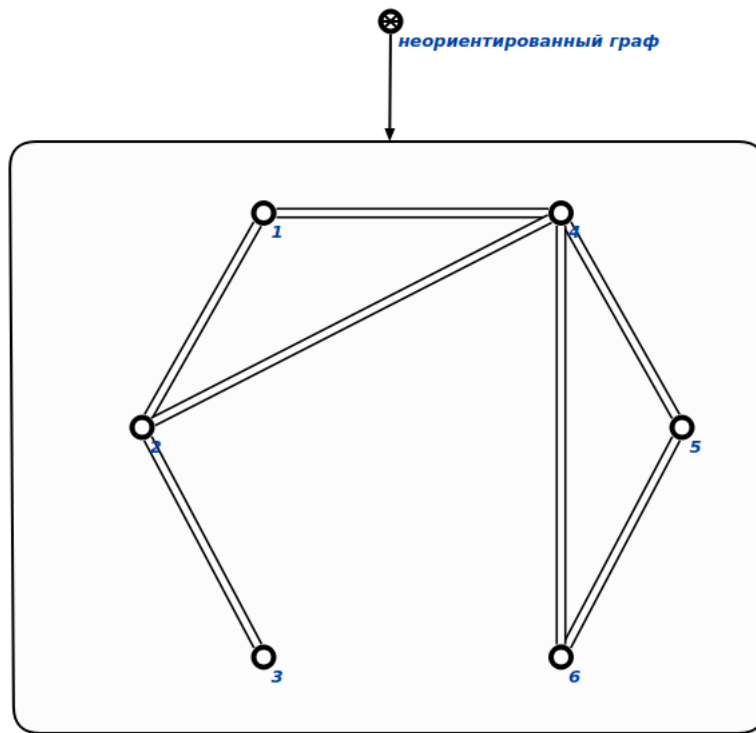


Рис. 19: Вход теста 5

Выход:

Будут найдены вершины 2, 4:

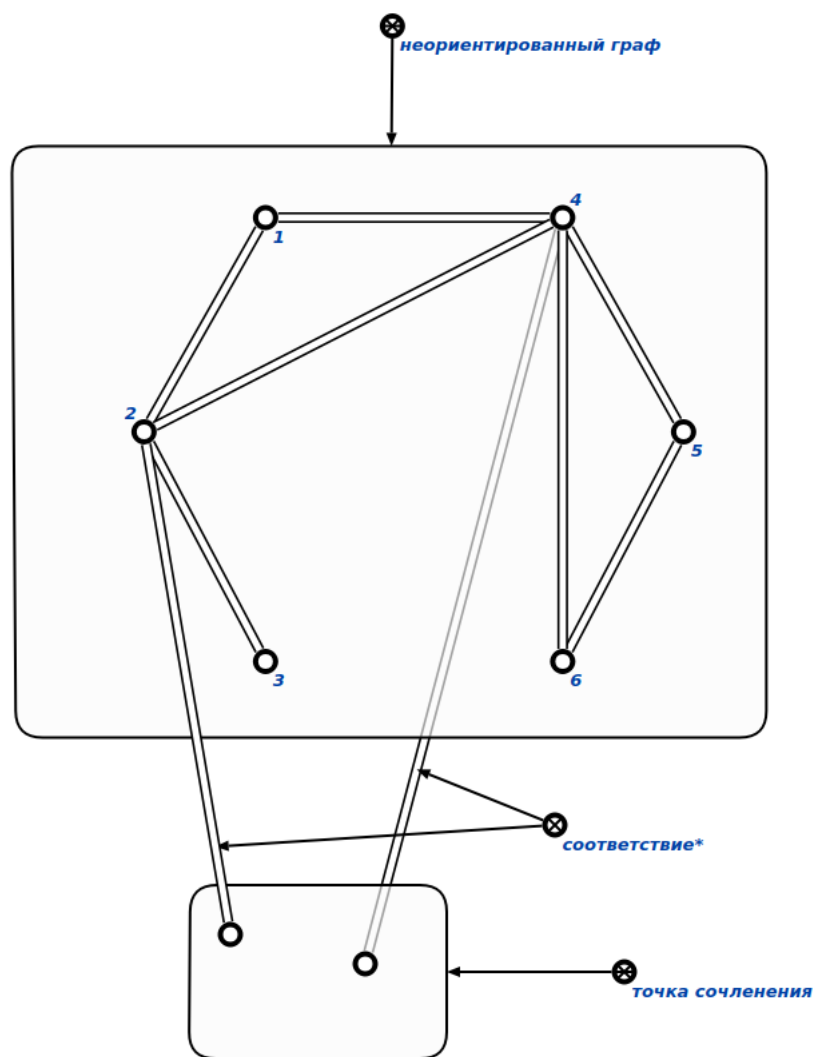
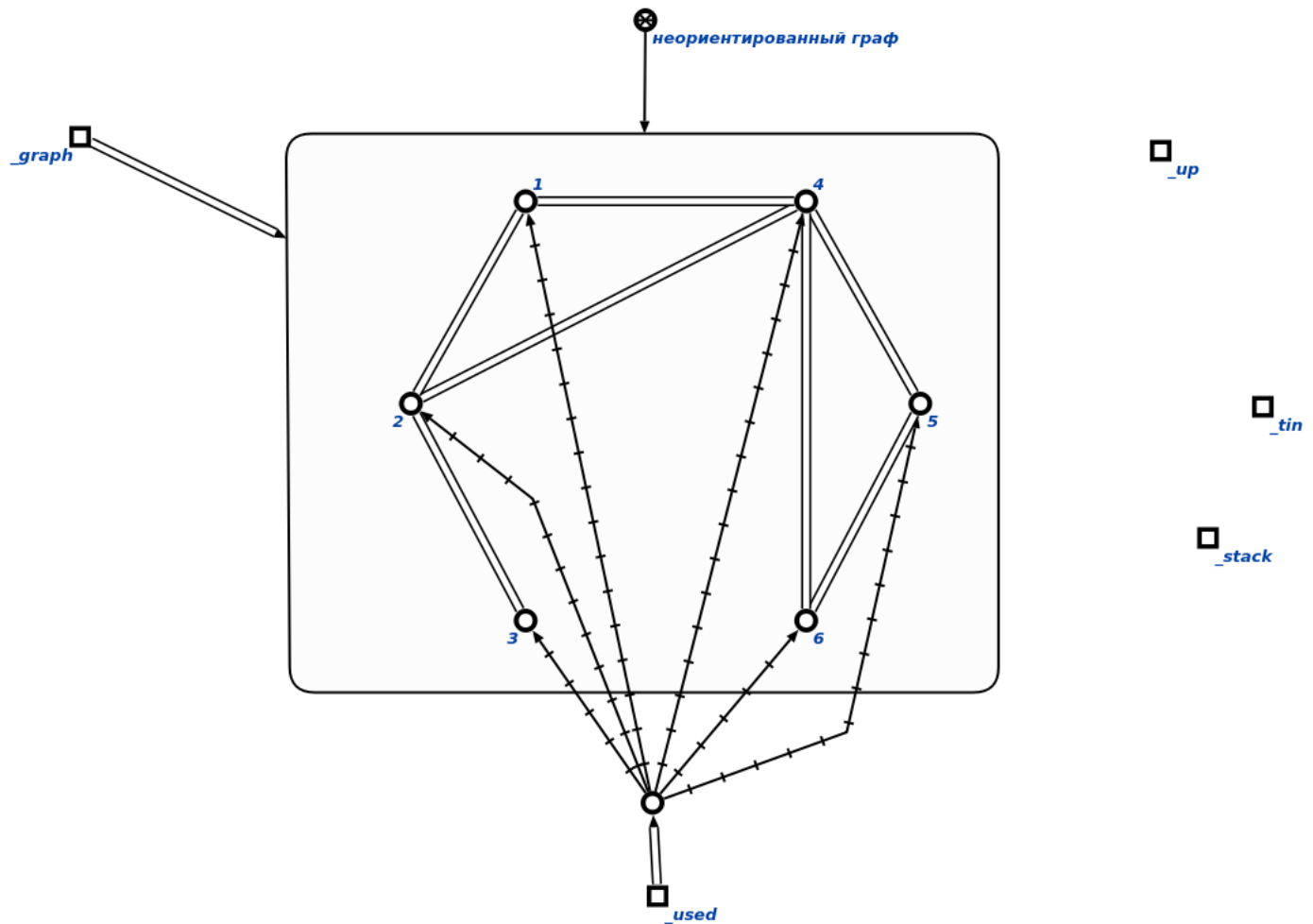


Рис. 20: Выход теста 5

3 ПРИМЕР РАБОТЫ АЛГОРИТМА В СЕМАНТИЧЕСКОЙ ПАМЯТИ

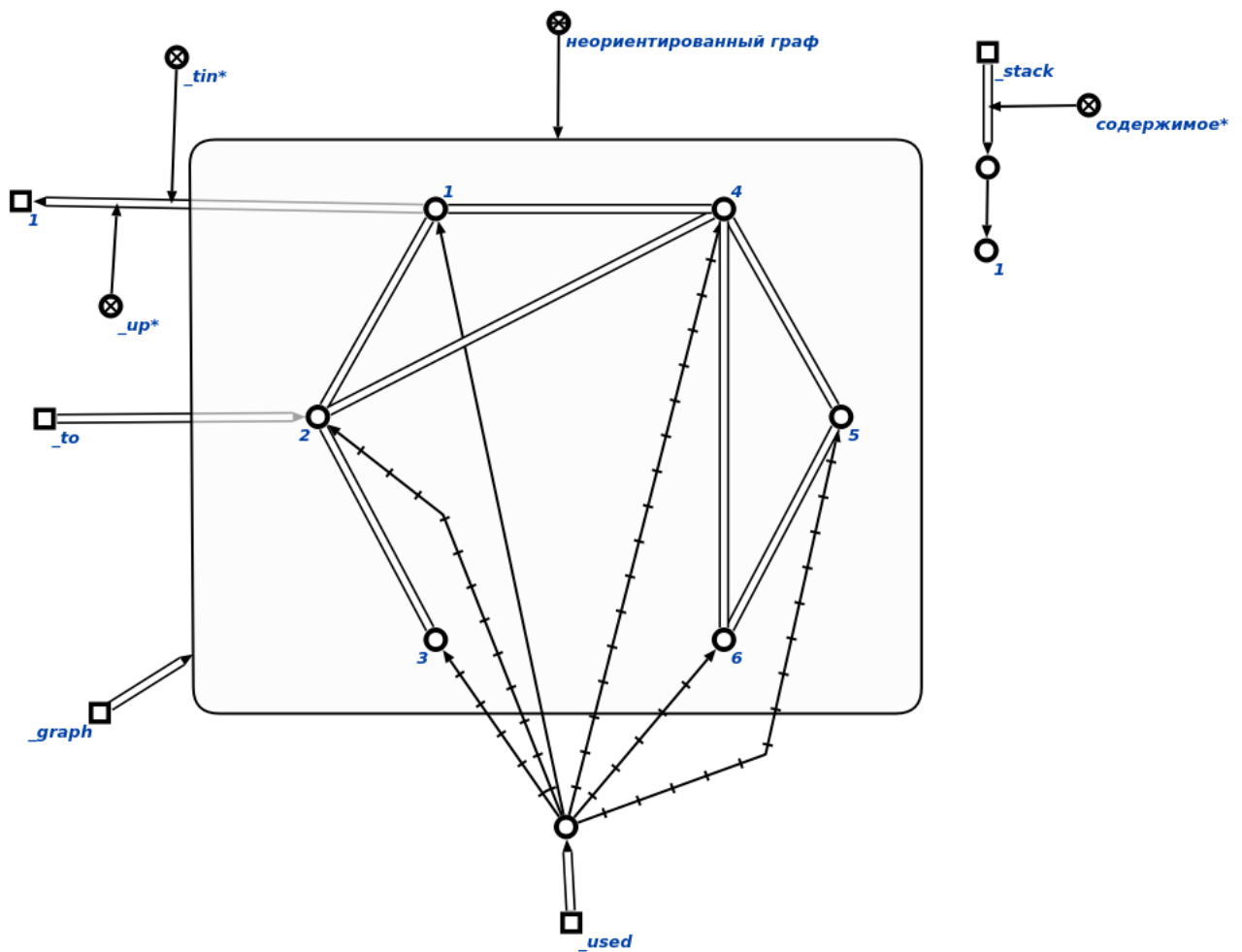
Создание графа с начальными переменными (Шаг 1)



`_graph` получит в качестве значения sc-узел неориентированного графа

`_used` - множество, не содержащее элементов

Начинаем поиск в глубину из вершины 1 (Шаг 2)



В `_used` добавляем вершину 1

`_tin` - время входа поиска в глубину в некоторую вершину

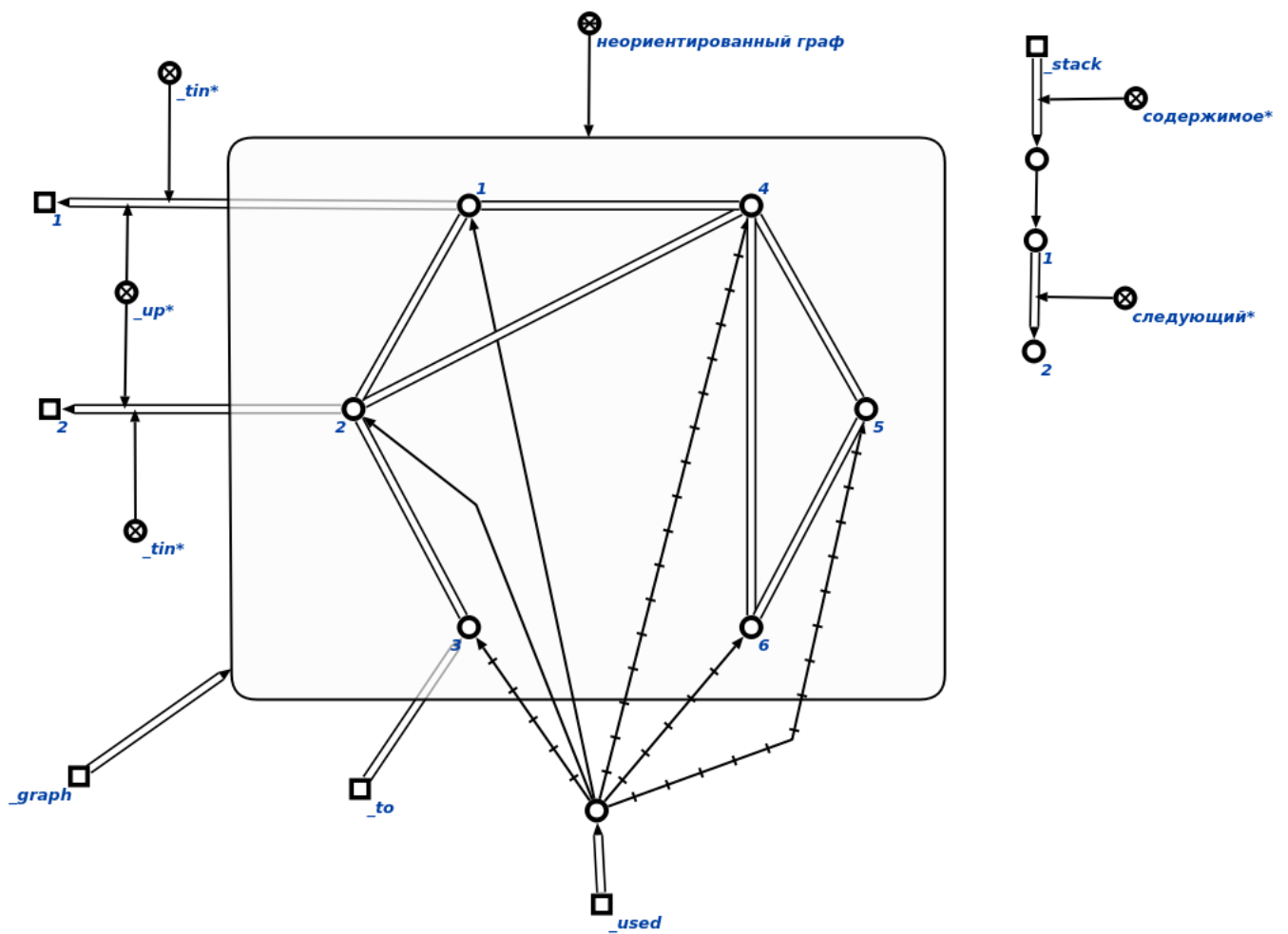
`_up` - время выхода поиска в глубину из некоторой вершины

`_to` - вершина, в которую двигается поиск в глубину

`_stack` - список вершин, просмотренных поиском в глубину для какой-либо вершины

В `_stack` добавляем вершину 1

Начинаем поиск в глубину из вершины 2 (Шаг 3)

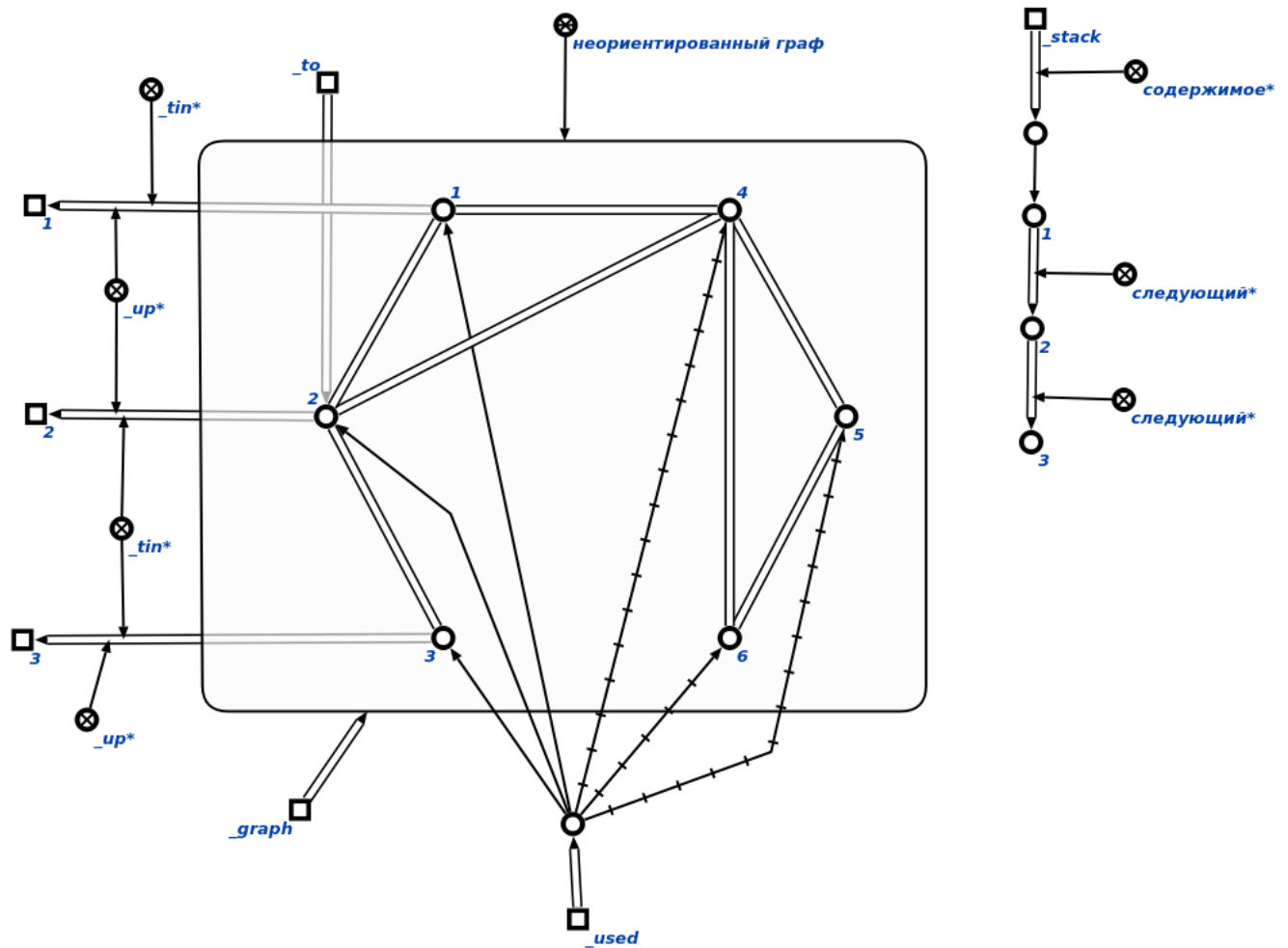


В `_used` добавляем вершину 2

`_to` получает вершину 3

В `_stack` добавляем вершину 2

Начинаем поиск в глубину из вершины 3 (Шаг 4)



В `_used` добавляем вершину 3

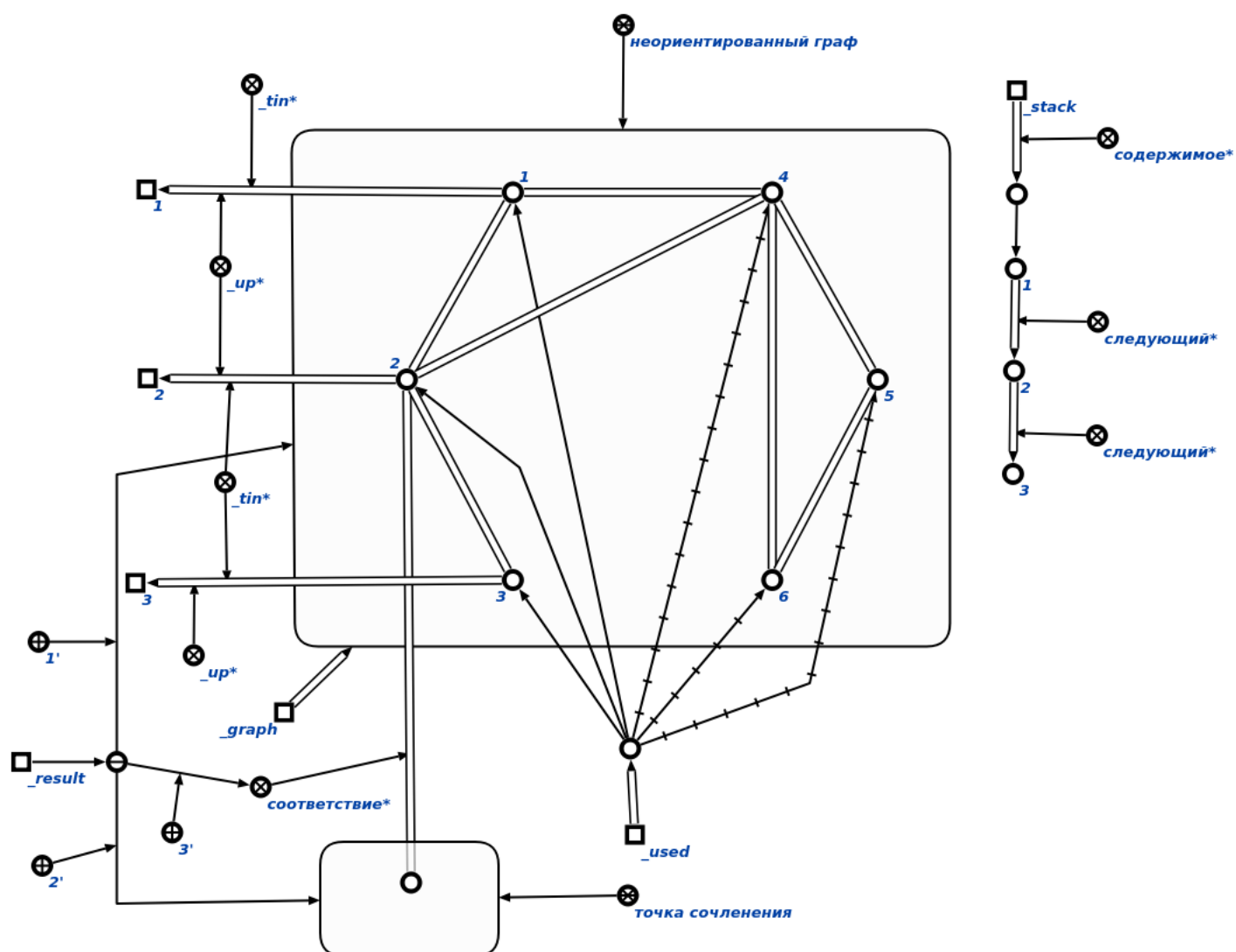
`_to` получает вершину 2

В `_stack` добавляем вершину 3

Поскольку вершина 2 уже была просмотрена и вершина 3 не имеет других смежных вершин, пересчитываем время выхода из неё.

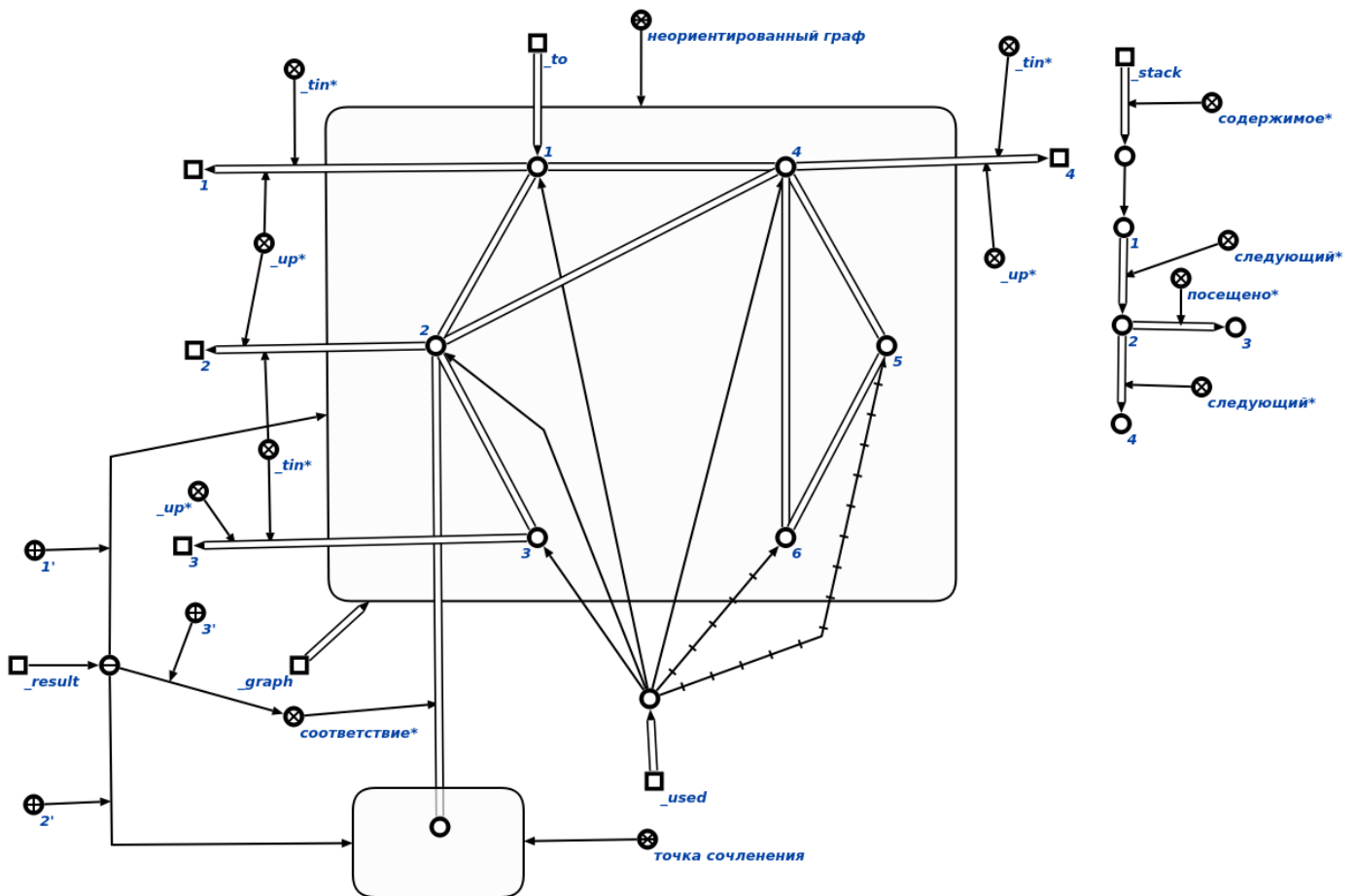
Т.к. выход из вершины 3 произошёл после входа в вершину 2, то вершина 2 является точкой сочленения.

Добавляем вершину 2 в множество точек сочленения (Шаг 5)



Создаём переменную _result , которая в качестве значения получит безымянную связку

Возвращаемся к вершине 2 и начинаем поиск в глубину из вершины 4 (Шаг 6)

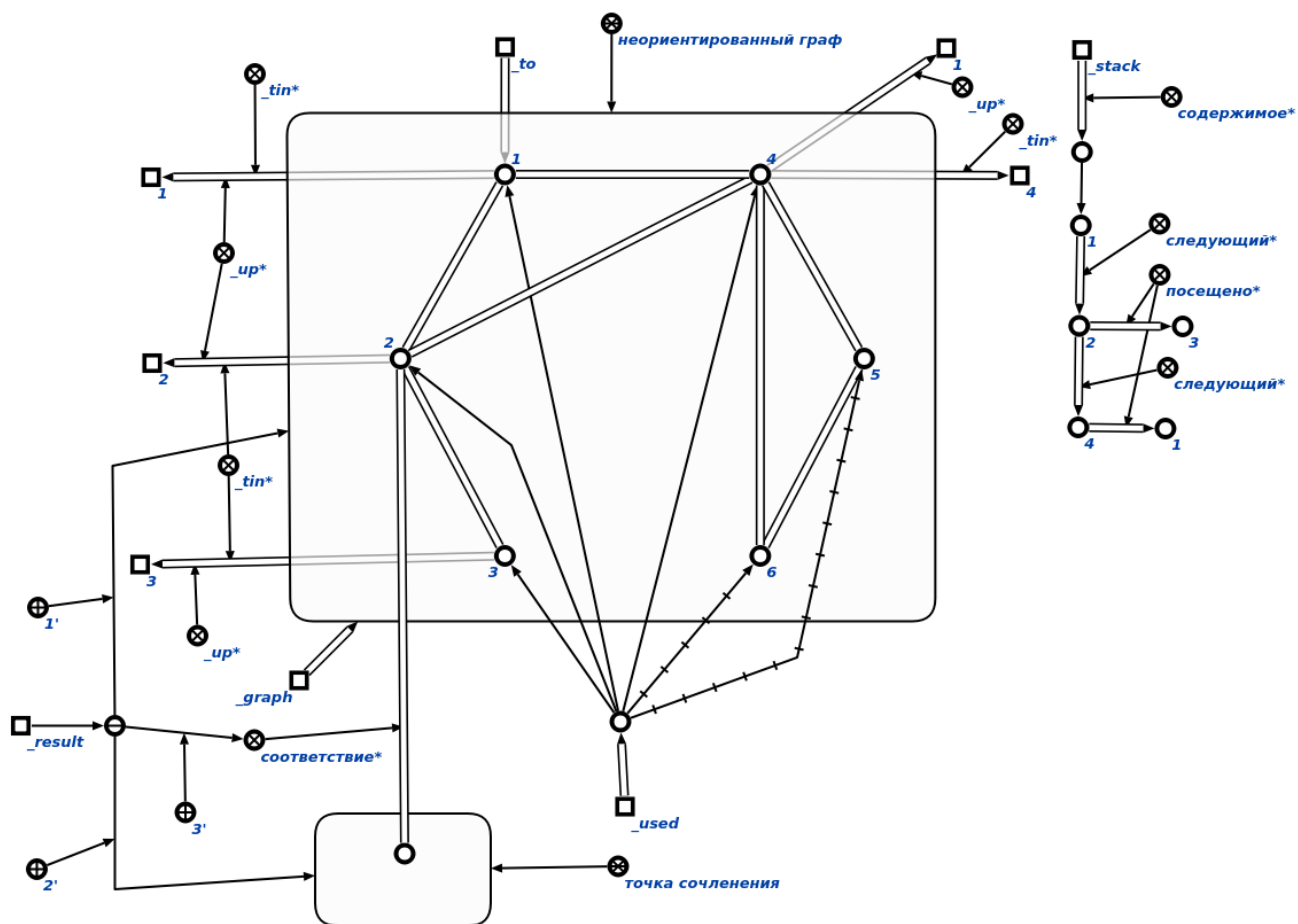


В `_stack` переопределяем отношение между вершинами 2 и 3 (*следующий** заменяем на *посещено**)

Добавляем вершину 4 в `_stack`

В `_used` добавляем вершину 4

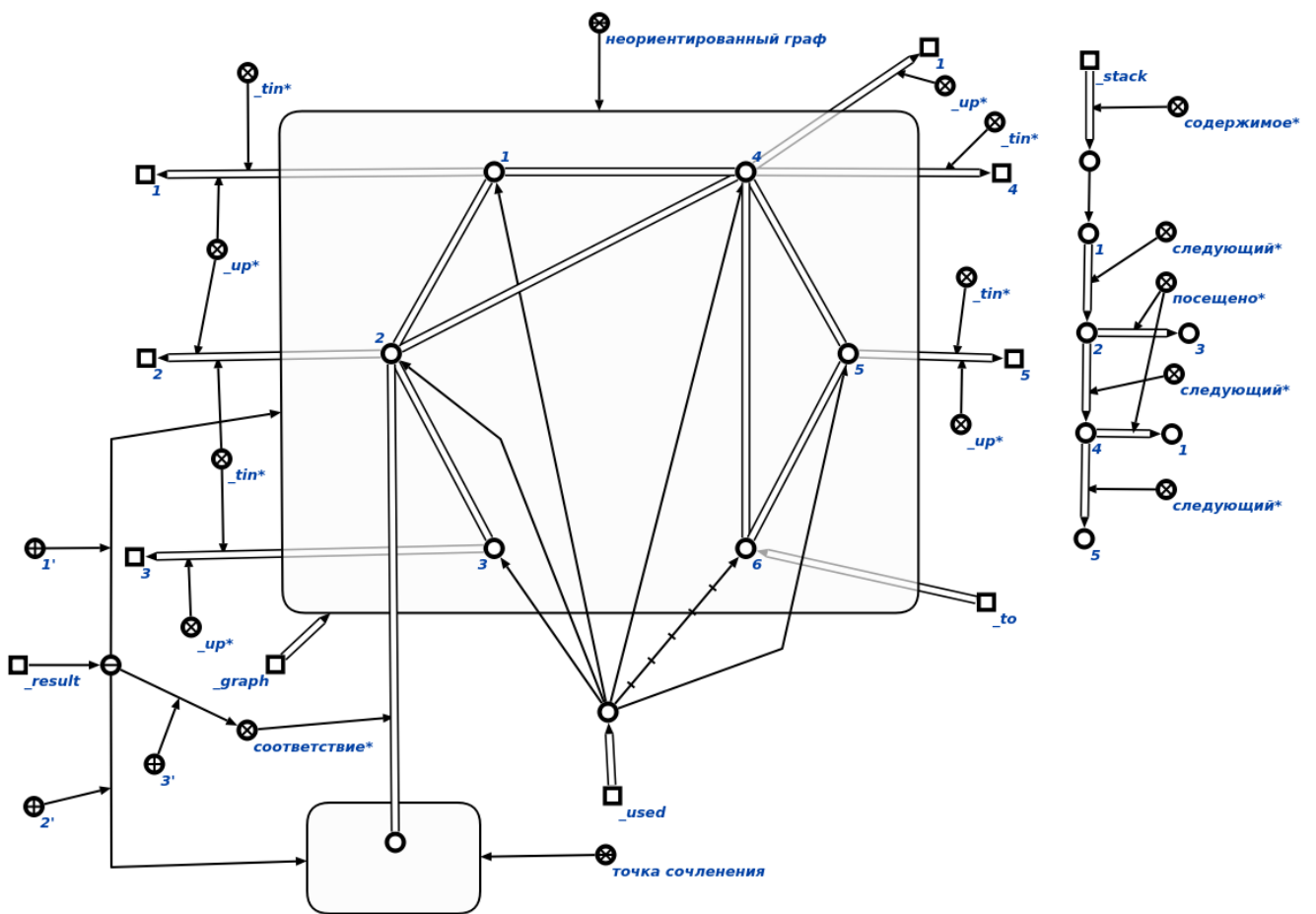
`_to` получает вершину 1



Поскольку вершина 1 уже была просмотрена, то пересчитываем время выхода из вершины 4

Отмечаем вершину 1 просмотренной при поиске в глубину из вершины 4

Начинаем поиск в глубину из вершины 5 (Шаг 7)

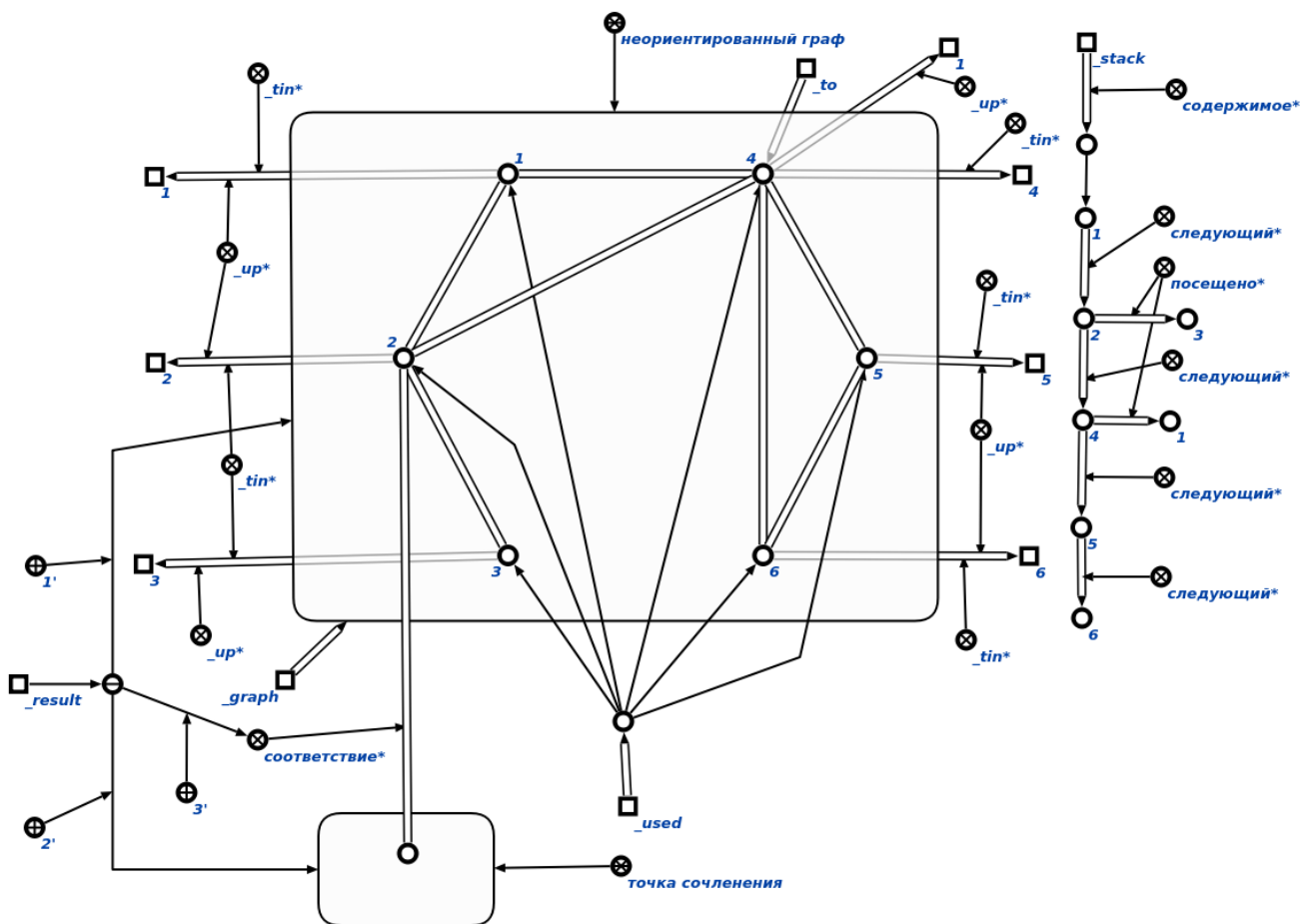


Добавляем вершину 5 в `_stack`

В `_used` добавляем вершину 5

__to получает вершину 6

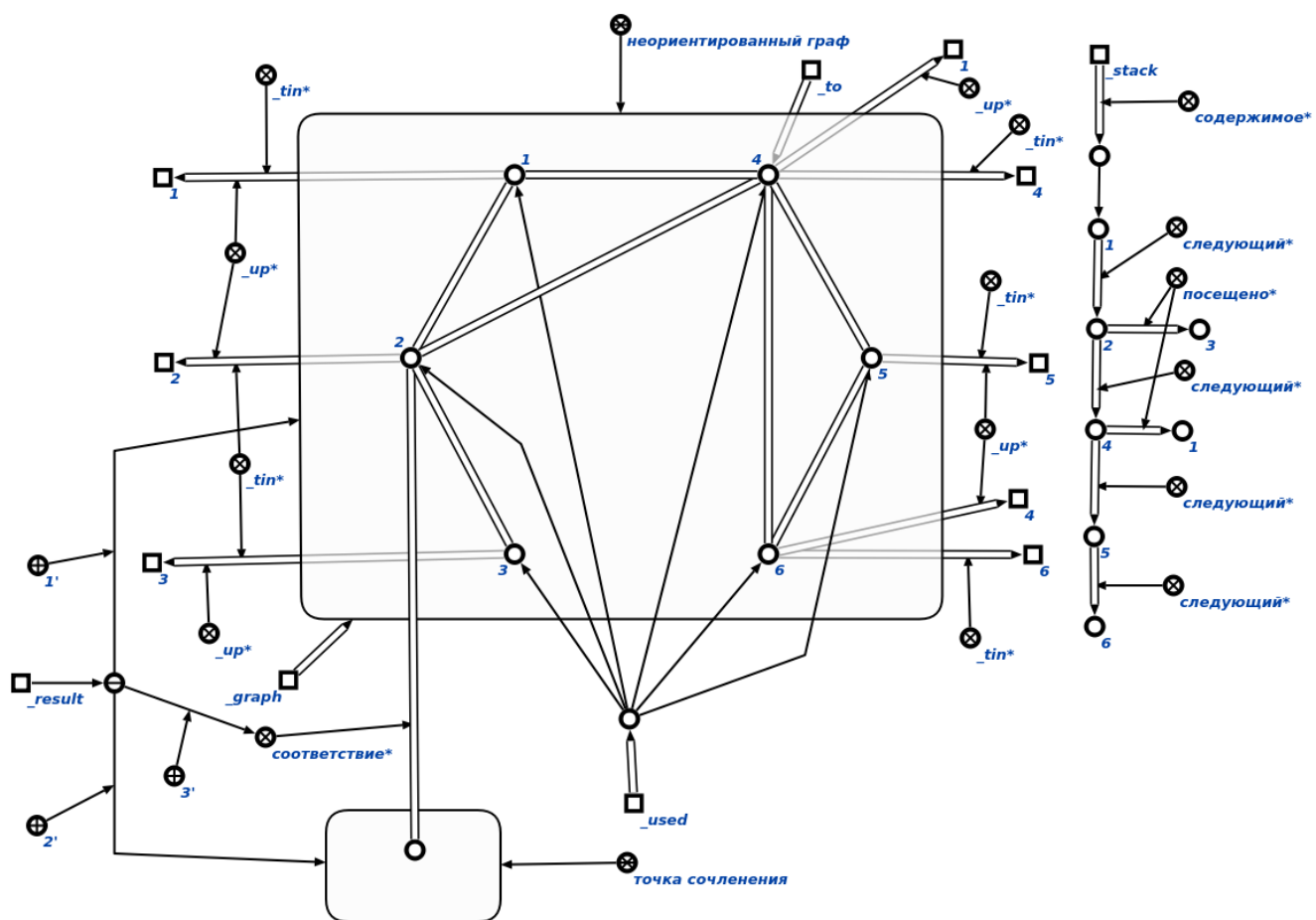
Начинаем поиск в глубину из вершины 6 (Шаг 8)



В `_stack` добавляем вершину 6

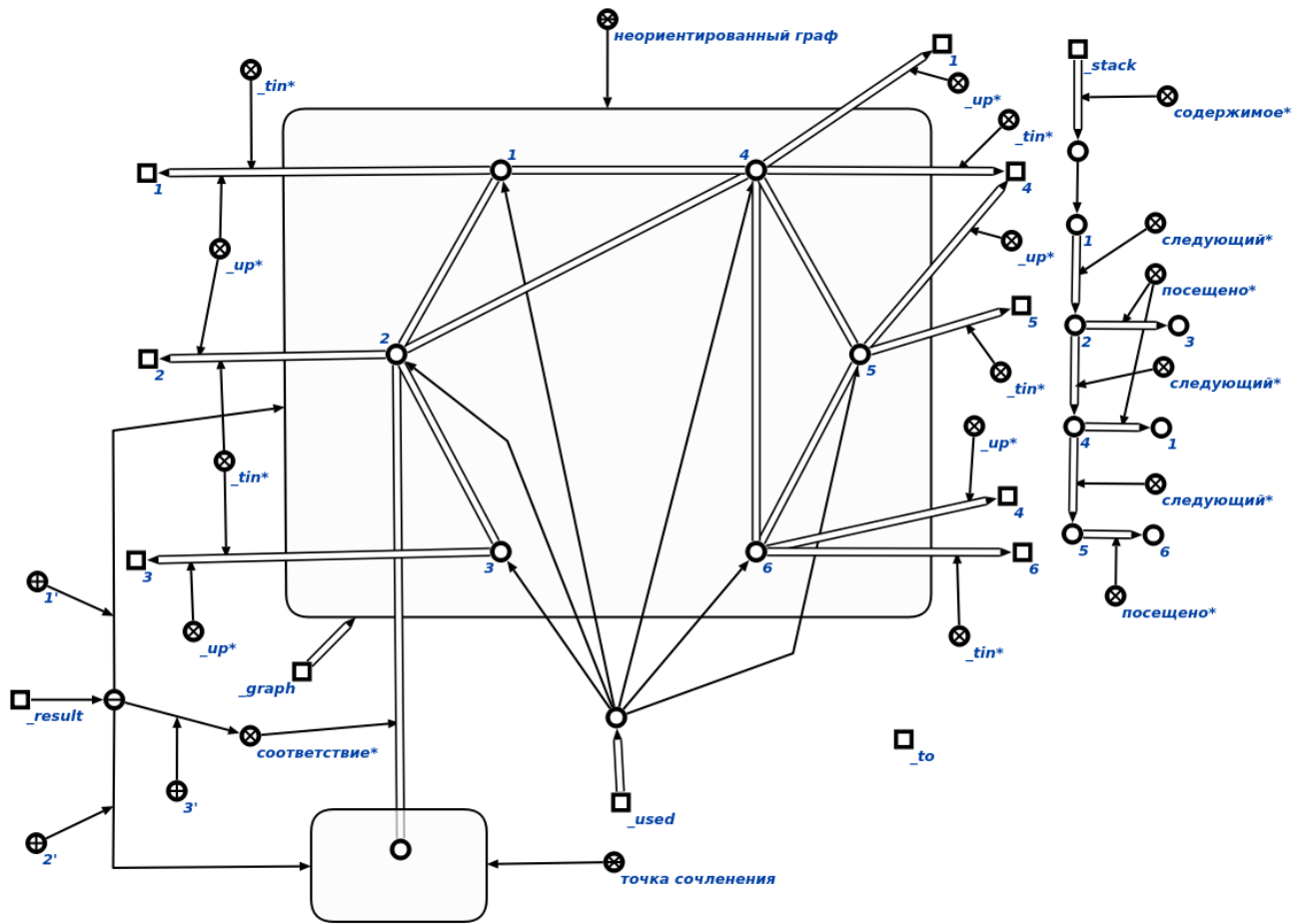
В `_used` добавляем вершину 6

_to получает вершину 4



Поскольку вершина 4 уже была просмотрена, то пересчитываем время выхода из вершины 6

Возвращаемся к вершине 5 (Шаг 9)



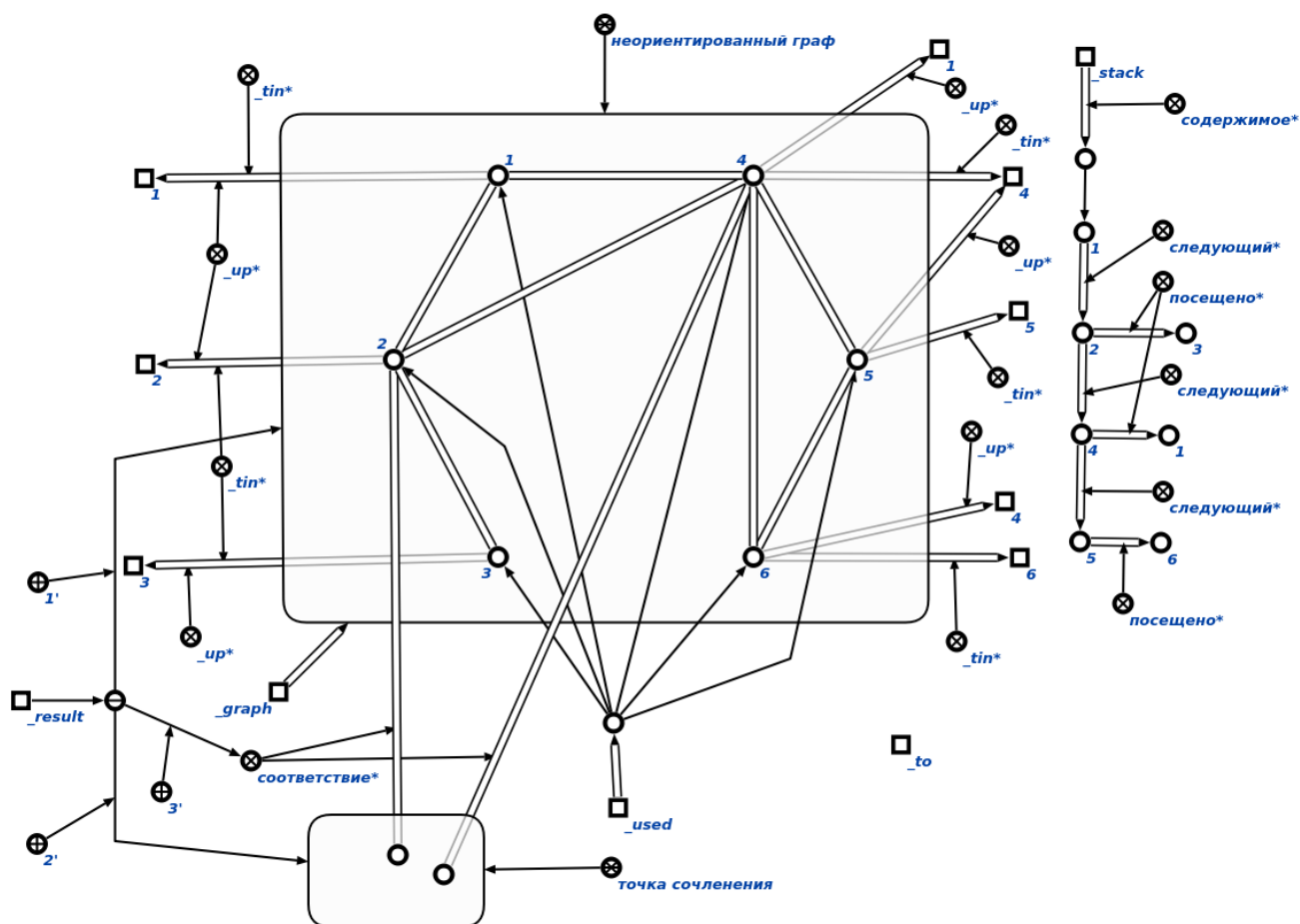
Поскольку все вершины смежные с вершиной 6 уже просмотрены, пересчитываем время выхода из вершины 5

Т.к. выход из вершины 6 произошёл раньше входа в вершину 5, вершина 5 не является точкой сочленения

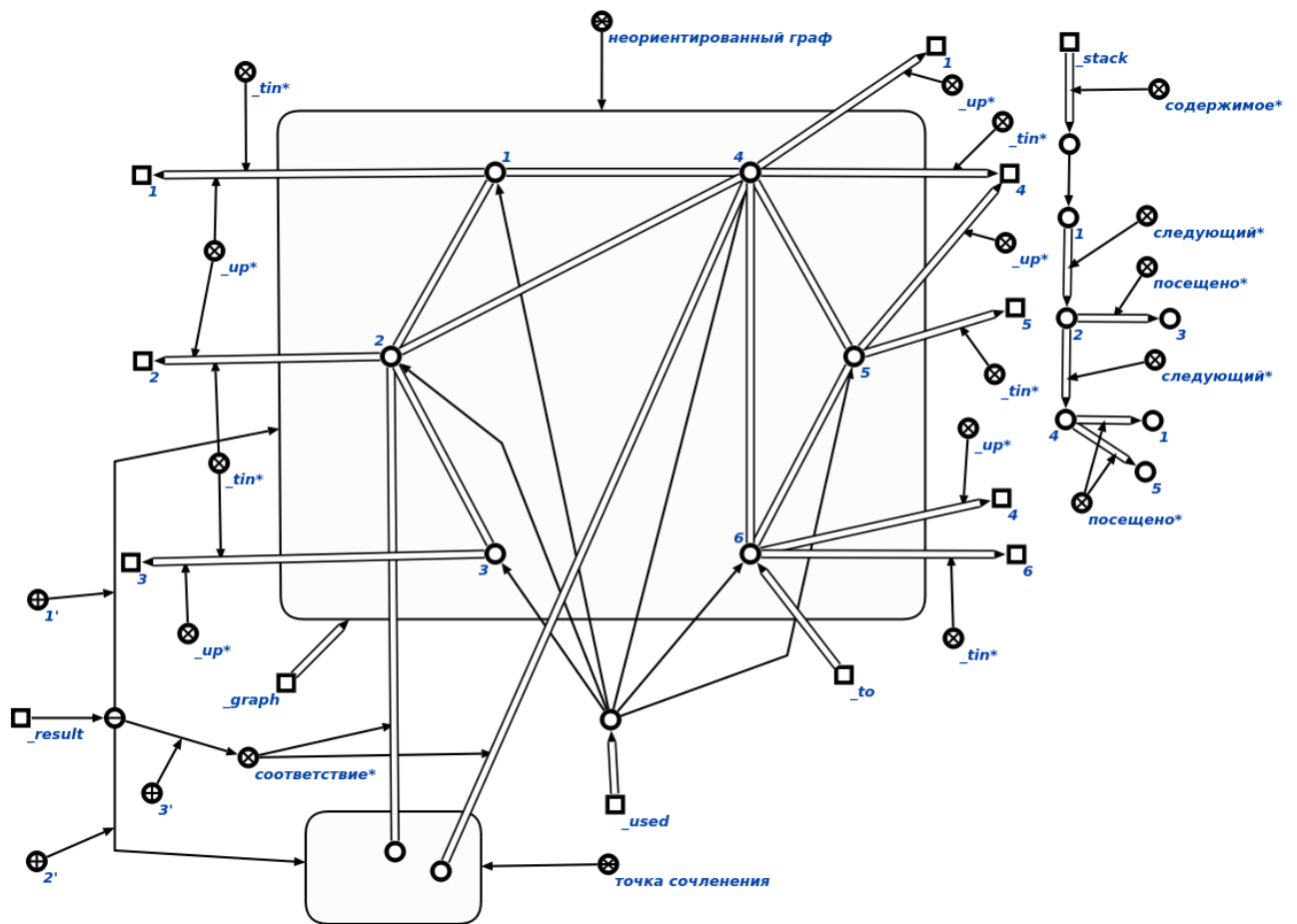
Т.к. выход из вершины 5 произошёл одновременно с входом в вершину 4, вершина 4 является точкой сочленения

В `_stack` переопределяем отношение между вершинами 5 и 6 (*следующий** заменяем на *посещено**)

Добавляем вершину 4 в множество точек сочленения (Шаг 10)



Возвращаемся к вершине 4 (Шаг 11)

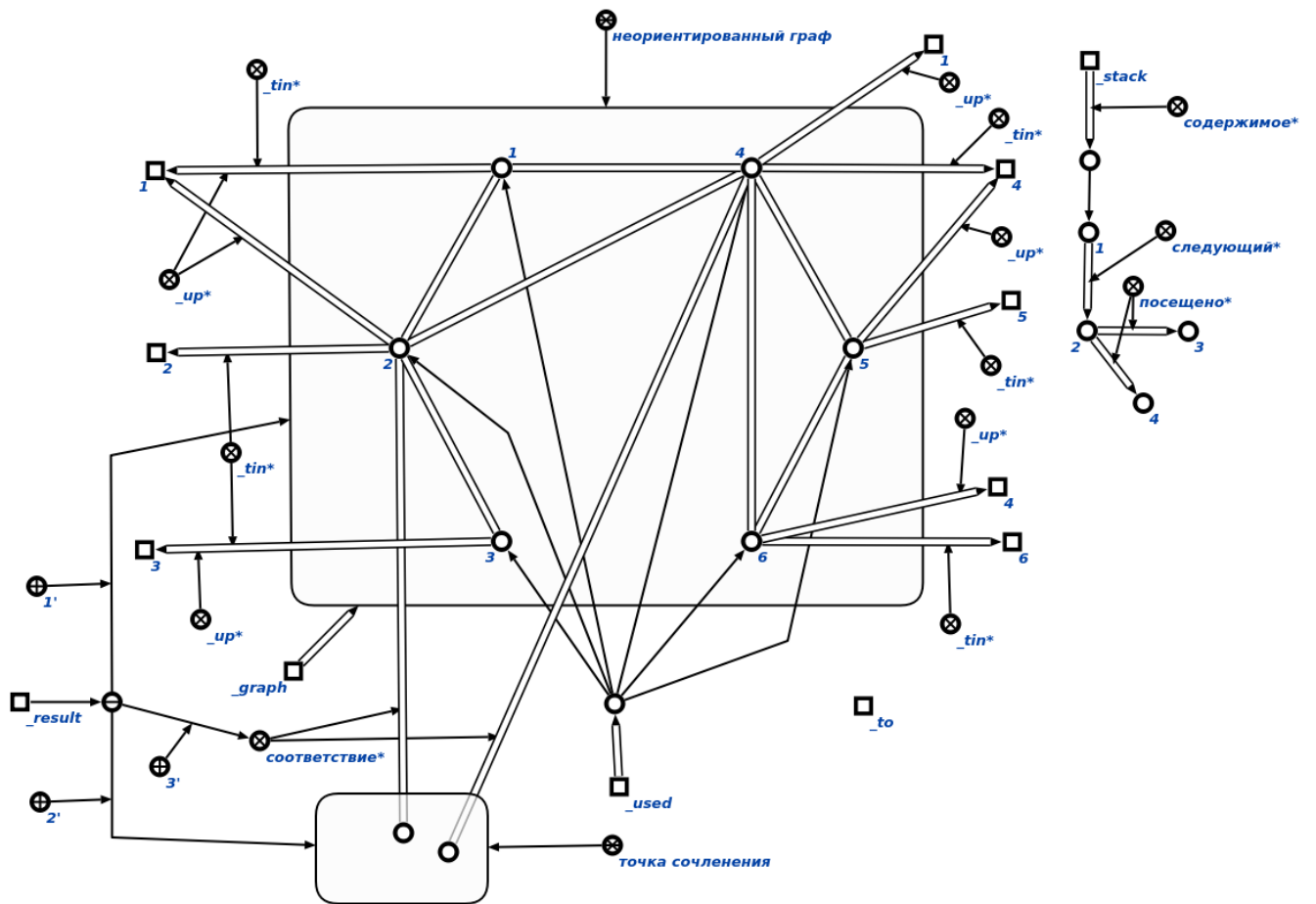


В `_stack` переопределяем отношение между вершинами 4 и 5 (*следующий** заменяем на *посещено**)

Поскольку вершина 6 уже была просмотрена, пересчитываем время выхода из вершины 4

Т.к.

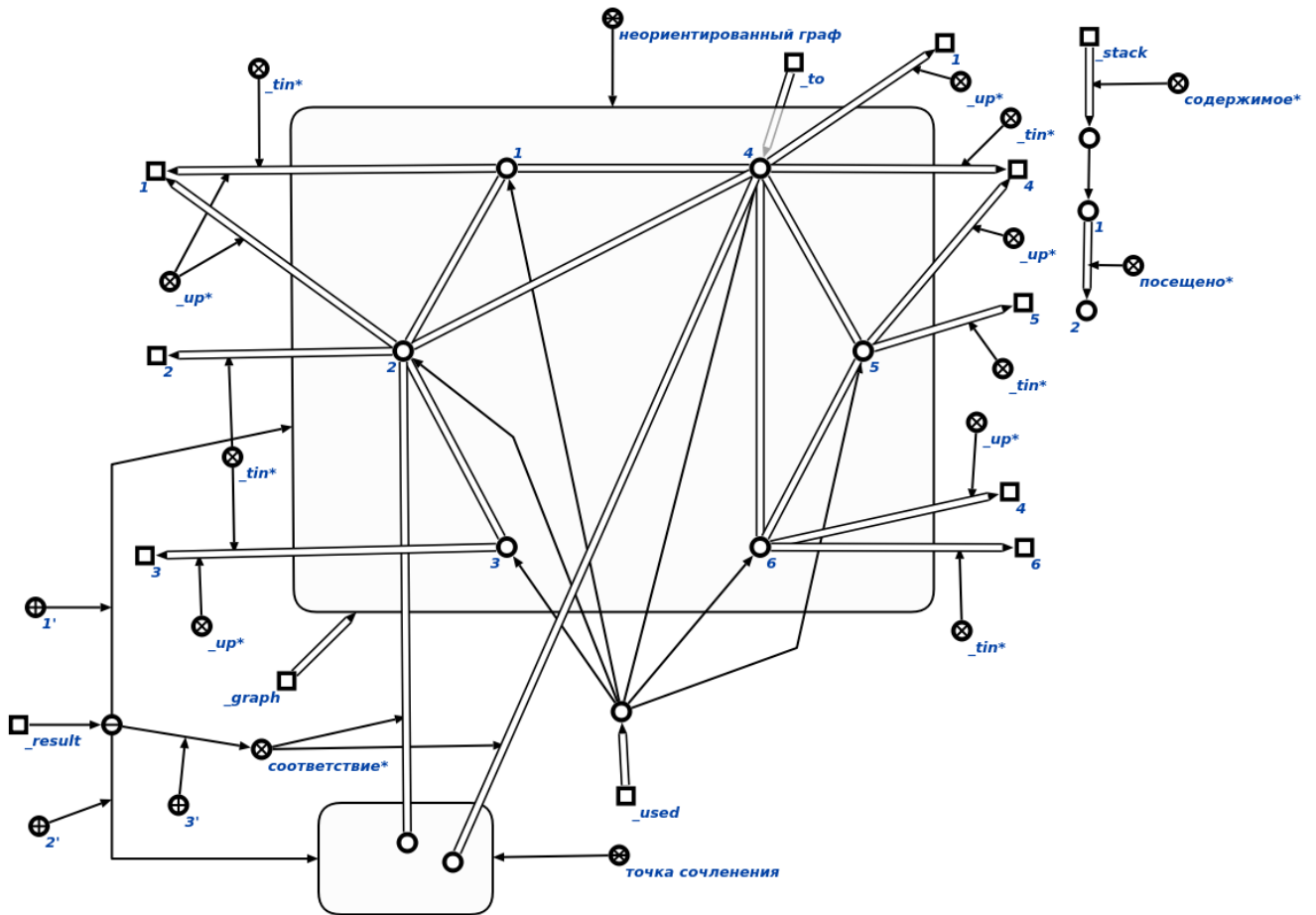
Возвращаемся к вершине 2 (Шаг 12)



В `_stack` переопределяем отношение между вершинами 2 и 4 (*следующий** заменяем на *посещено**)

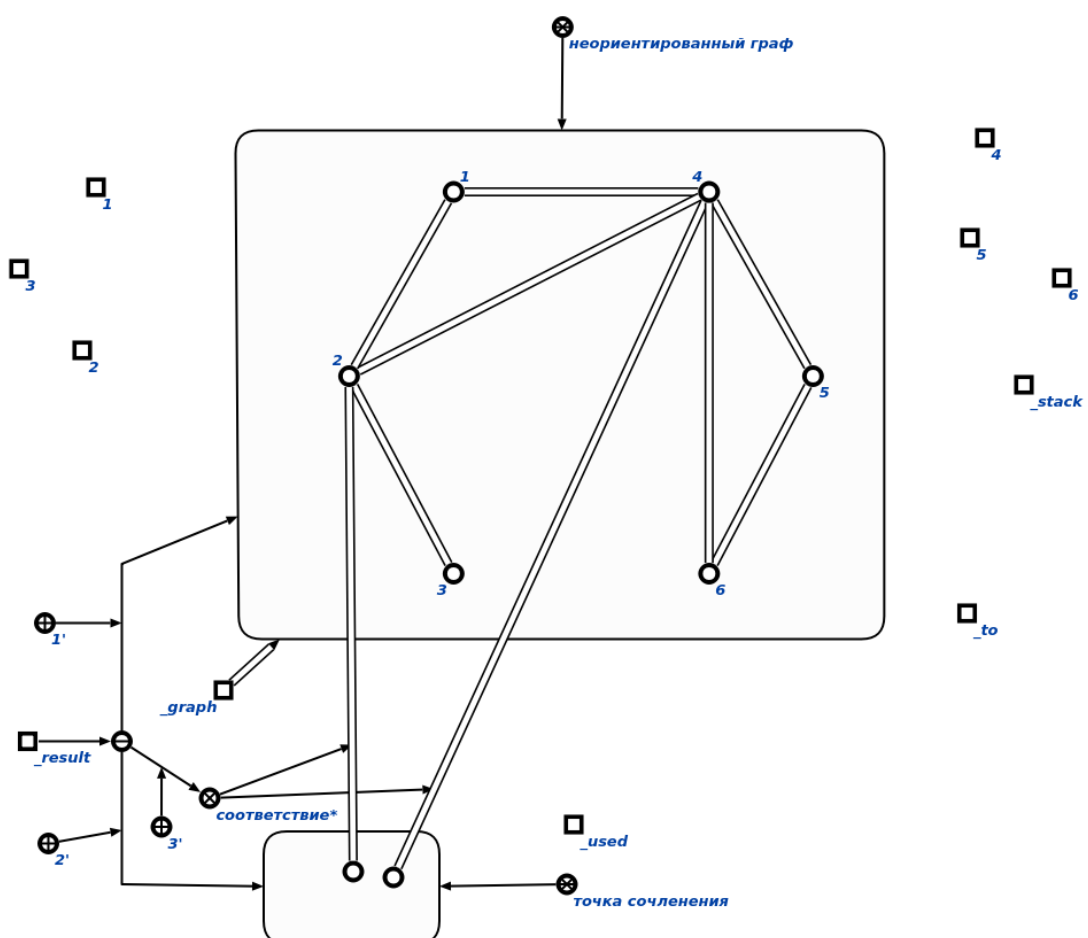
Поскольку все вершины смежные с вершиной 2 уже просмотрены, пересчитываем время выхода из вершины 2

Возвращаемся к вершине 1 (Шаг 13)

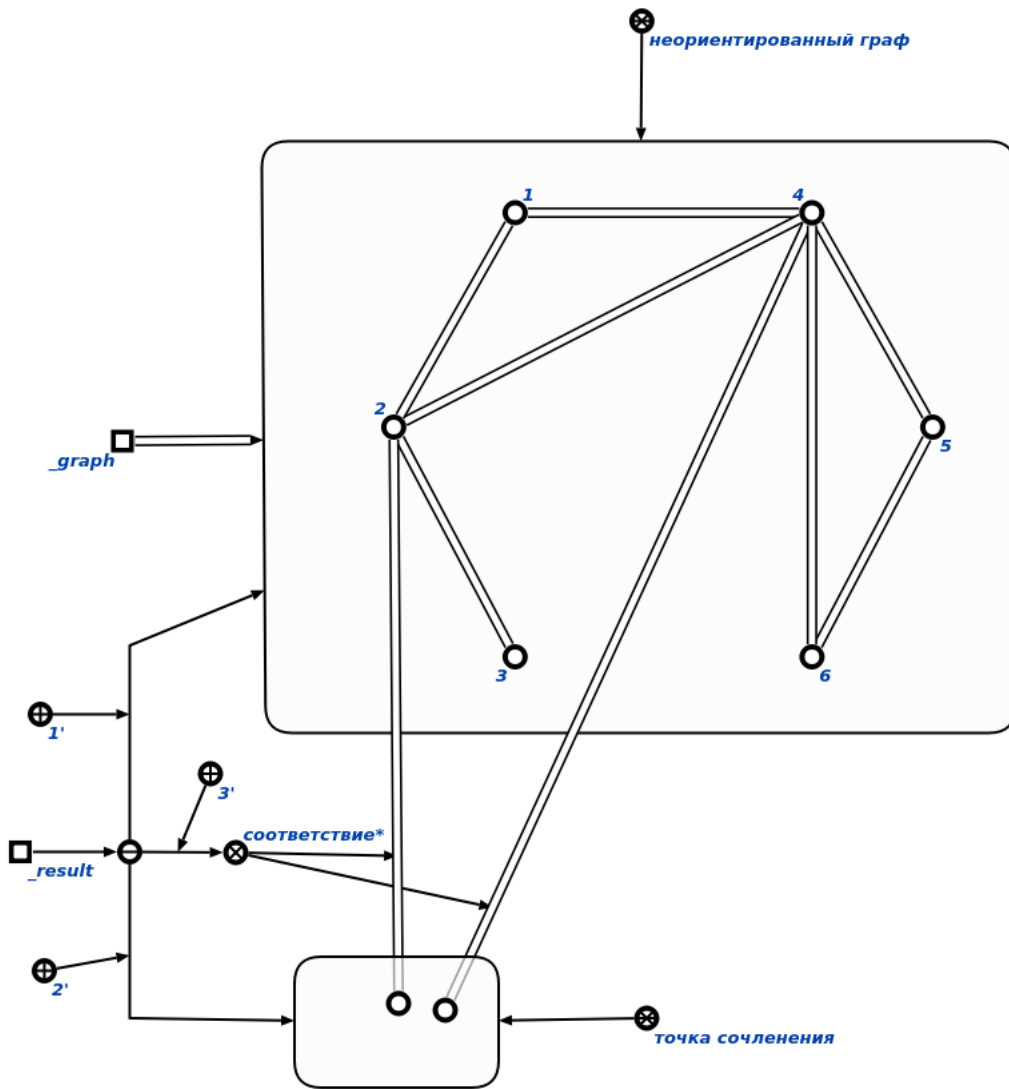


Поскольку вершина 4 уже была просмотрена, пересчитываем время выхода из вершины 1

Удаление ненужных связей (Шаг 14)



Результат работы алгоритма



На данном этапе продемонстрирован результат работы алгоритма, значение переменной `_result` будет возвращено в вызывающий контекст.

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.