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

Найти минимальное и среднее расстояние между периферийными вершинами неориентированного графа.

Выполнил: Е. А. Рублевская

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 минимальное и среднее расстояние между периферийными вершинами 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 существенен

   (a) Вершина (относительное понятие, ролевое отношение);

   (b) Связка (относительное понятие, ролевое отношение).

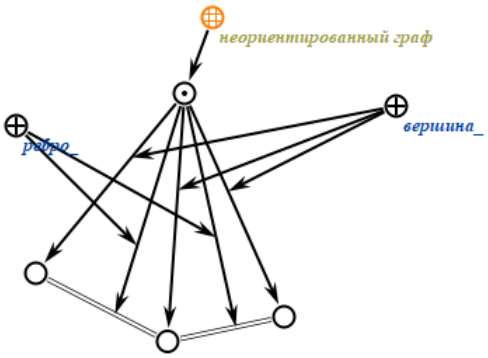

Рис. 1: Абсолютное понятие неориентированного графа

2. ***Эксцентриситетом*** вершины называется расстояние до самой дальней вершины графа

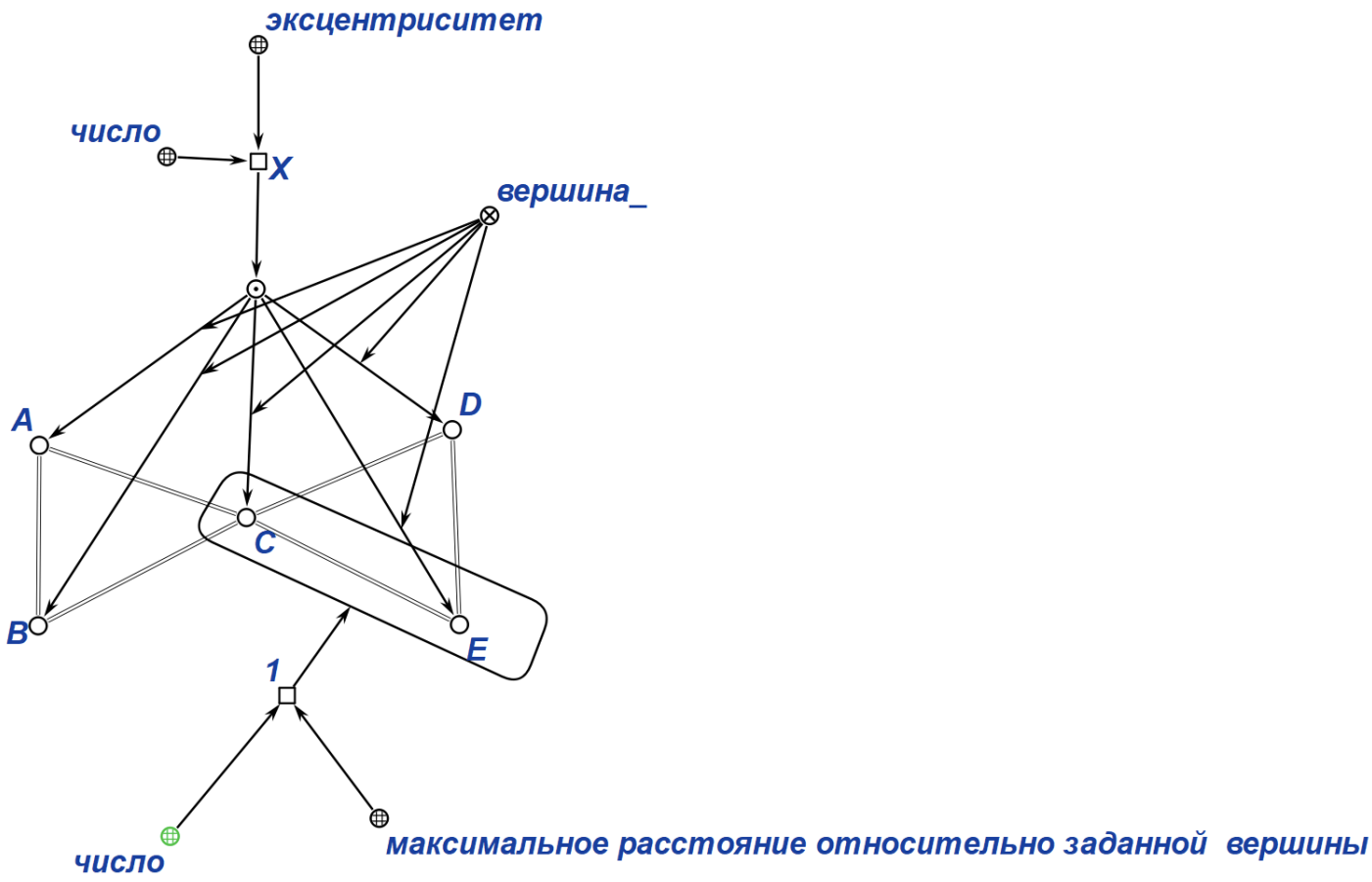

Рис. 2: Абсолютное понятие эксцентриситета

3. ***Диаметр графа*** — это максимальное расстояние между любыми двумя его вершинами.

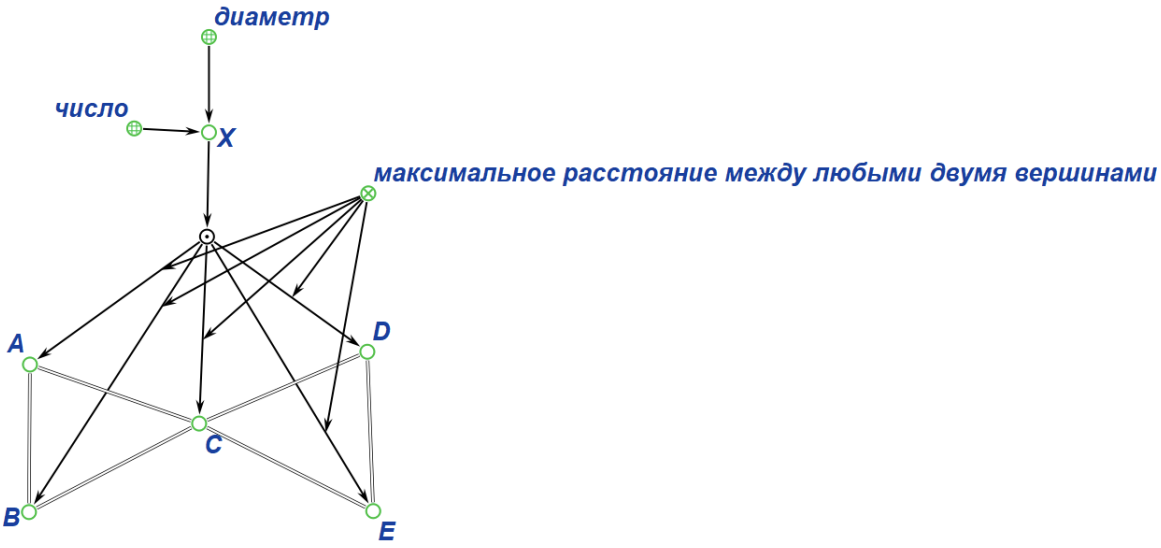

Рис. 3: Абсолютное понятие диаметра

4. ***Периферийная вершина*** — вершина, эксцентриситет которой равен диаметру графа.

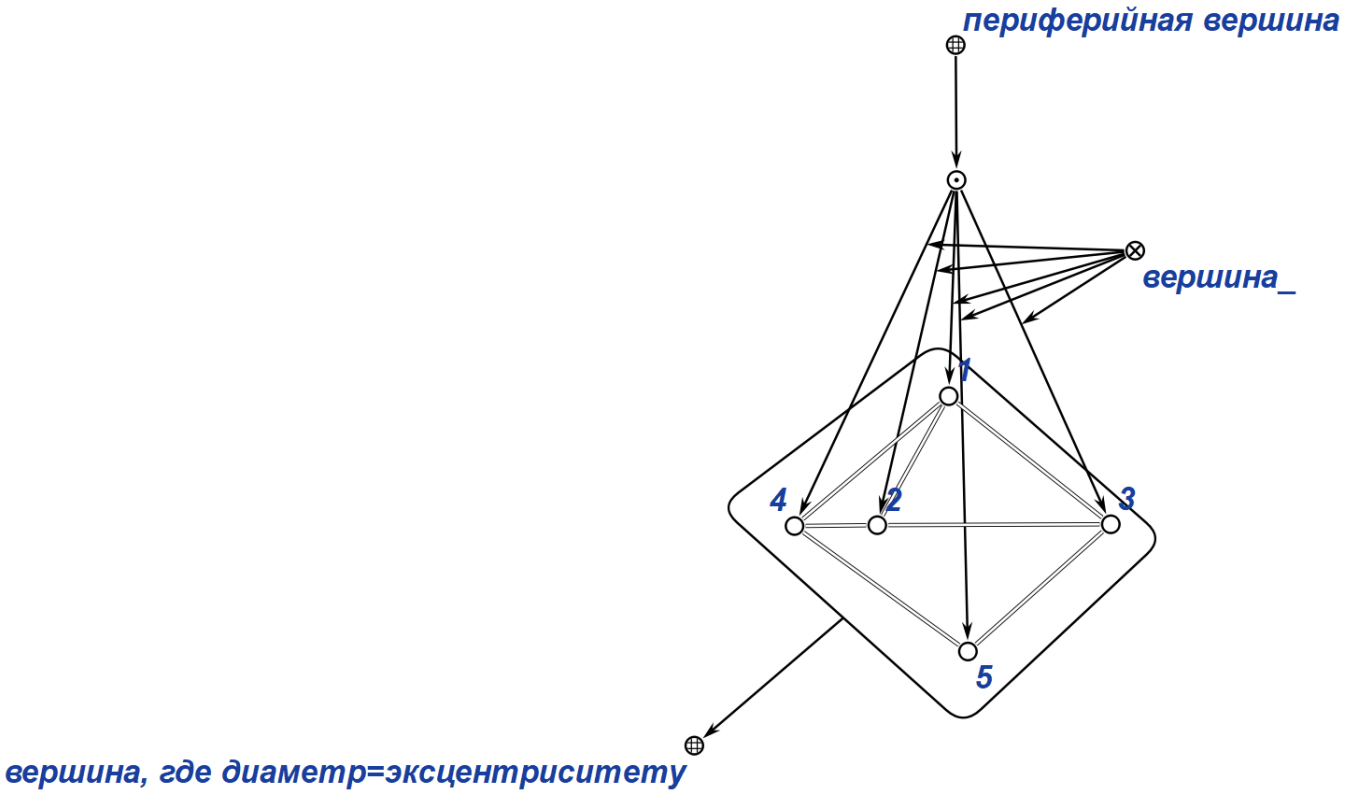

Рис. 4: Абсолютное понятие периферийной вершины

3 Тестовые примеры

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

3.1 Тест 1

Вход:

Необходимо найти минимальное и среднее расстояние между периферийными вершинами неориентированного графа.

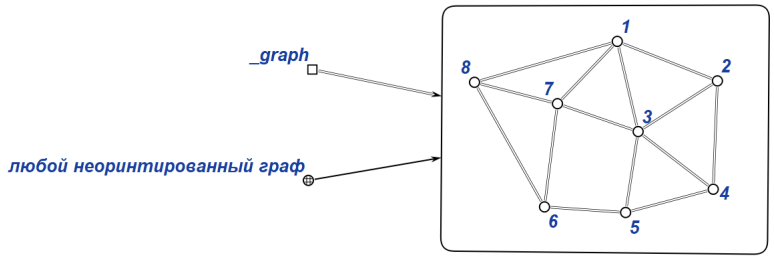

Рис. 5: Вход теста 1

Выход: Будет найдено два значения: минимальное (1) и среднее (1.8) расстояние между периферийными вершинами.

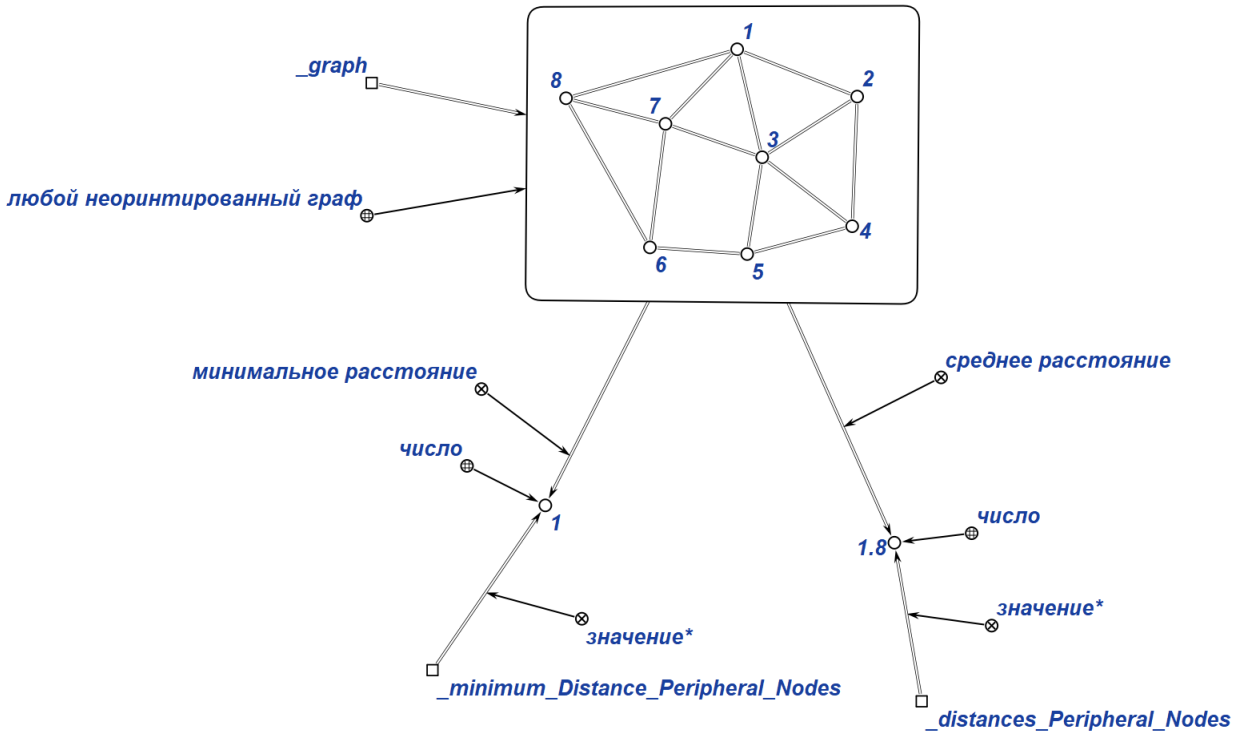

Рис. 6: Вход теста 1

3.2 Тест 2

Вход: Необходимо найти минимальное и среднее расстояние между периферийными вершинами неориентированного графа.

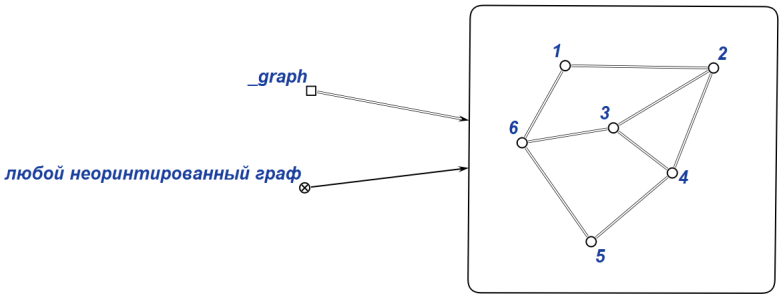

Рис. 7: Вход теста 2

Выход: Будет найдено два значения: минимальное (1) и среднее (1.5) расстояние между периферийными вершинами.

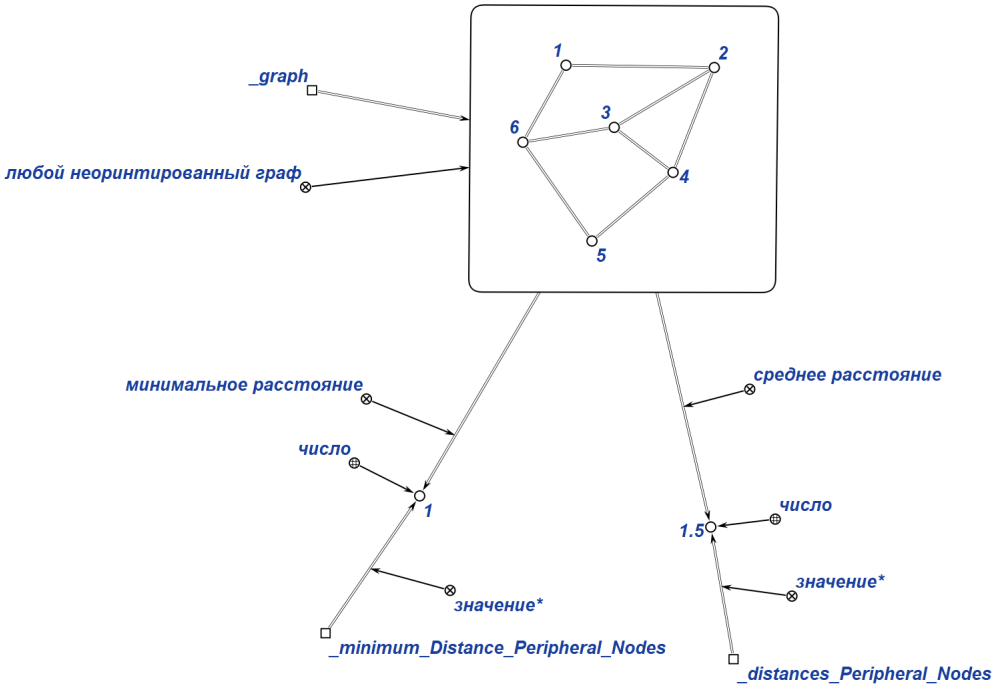

Рис. 8: Вход теста 2

3.3 Тест 3

Вход: Необходимо найти минимальное и среднее расстояние между периферийными вершинами неориентированного графа.

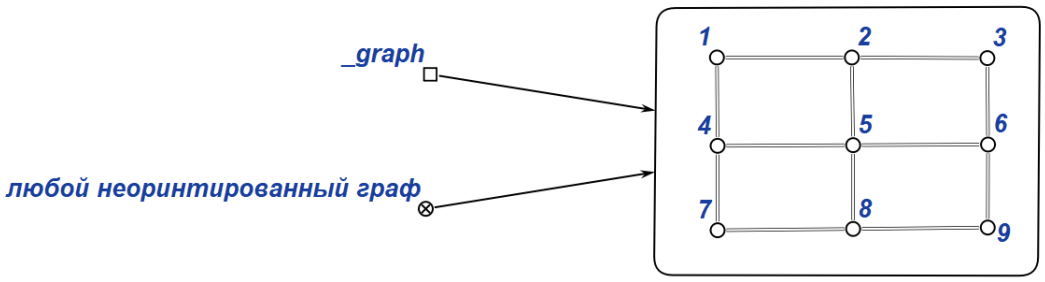

Рис. 9: Вход теста 3

Выход: Будет найдено два значения: минимальное (1) и среднее (2.4) расстояние между периферийными вершинами.

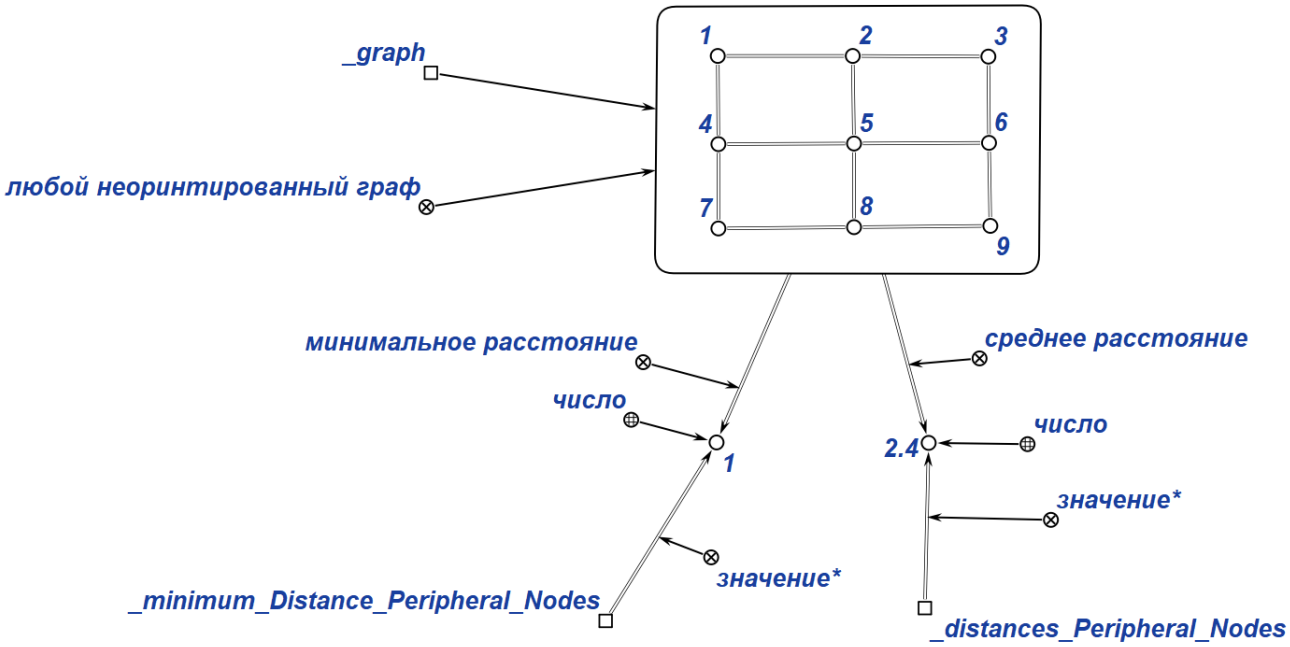

Рис. 10: Вход теста 3

3.4 Тест 4

Вход: Необходимо найти минимальное и среднее расстояние между периферийными вершинами неориентированного графа.

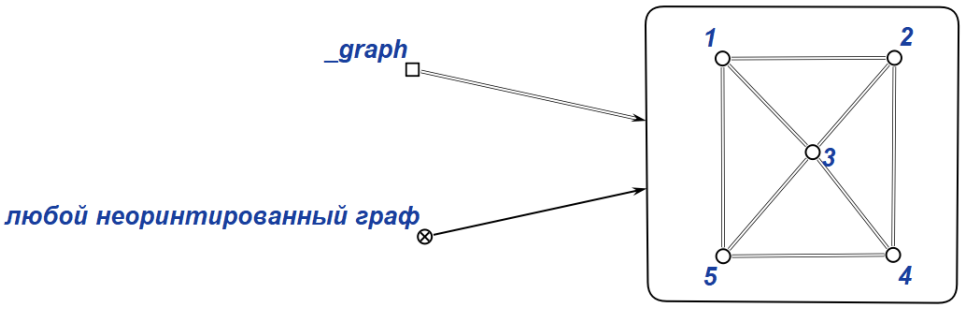

Рис. 11: Вход теста 4

Выход: Будет найдено два значения: минимальное (1) и среднее (1.3) расстояние между периферийными вершинами.

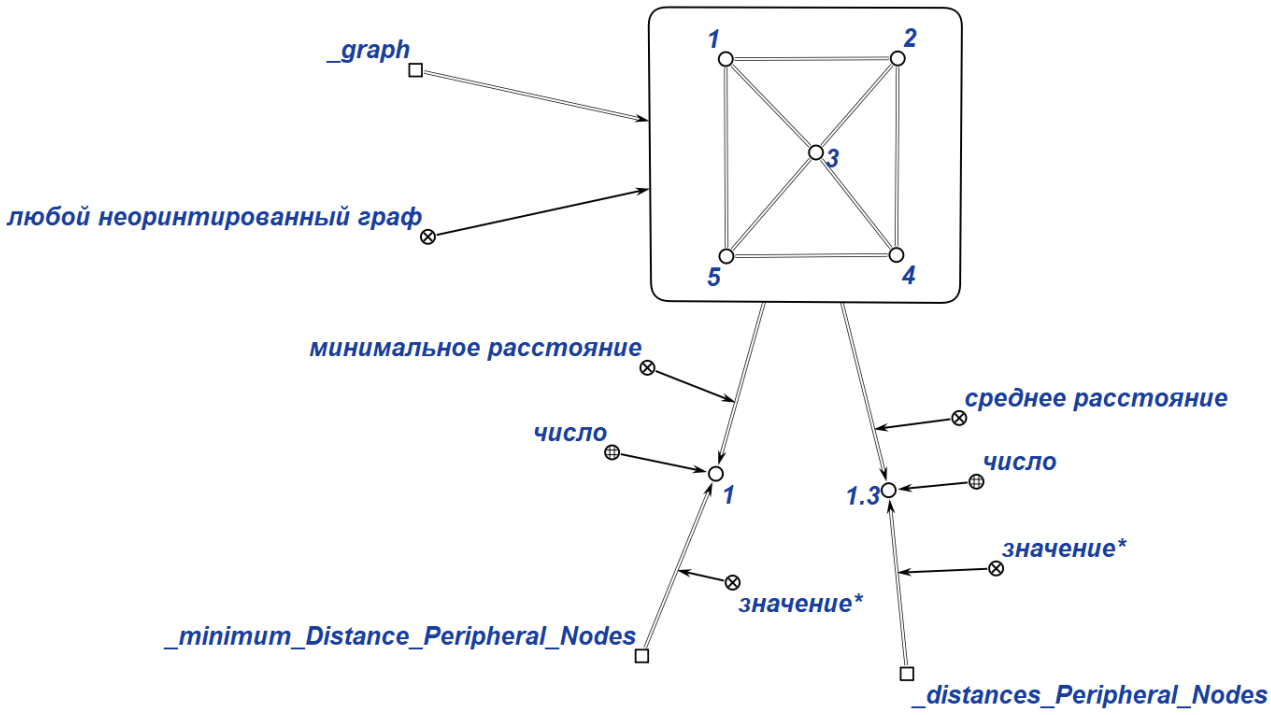

Рис. 12: Вход теста 4

4 Пример работы алгоритма в семантической памяти

4.1 Краткое описание:

1. первый элемент первого уровня содержит список
2. Добавить все вершины графа, кроме вершины, которую мы выбрали, во множество непосещенных вершин.
3. Создаем переменную (счетчик), в которой будет храниться эксцентриситет данной вершины.
4. Создать новую волну и добавить в нее вершину, которая еще не посещена.
5. При запуске новой волны счетчик будет увеличиваться на 1, если эксцентриситет увеличивается (если нет - счетчик не изменяется) .
6. Начальная волна - это новая волна. Новой волной будем называть последнюю созданную волну.
7. Сформировать следующую волну для новой волны. В нее попадют соседние вершины от заданной (далее от посещенных уже). Если вершина попала в формируемую волну, то ее надо исключить из множества непроверенных вершин. Созданную волну установить как следующую для новой волны, и после этого созданную волну считать новой волной.
8. Если новая волна пуста, то мы обошли весь граф. И получили значение эксцентриситетов всех вершин. Мы можем их сравнить и найти диаметр графа.
9. находим периферийные вершины
10. Запускаем еще одну волну (работает по принцепу, написанному ранее), но теперь волна находит расстояние между периферийными вершинами.
11. Находим минимальное расстояние между периферийными вершинами.
12. Находим среднее расстояние между периферийными вершинами.
13. Выводим полученный результат.

4.2 Демонстрация на тесте 5:

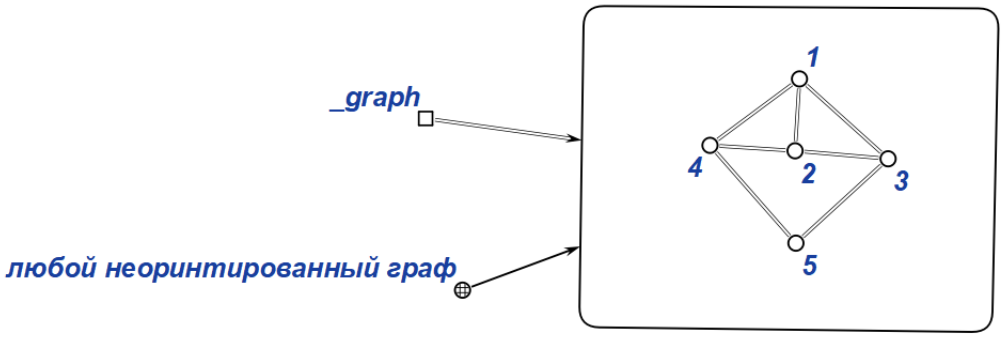

Рис. 13: Вход теста 5

1. ***graph*** получит в качестве значения sc-узел неориентированного графа;

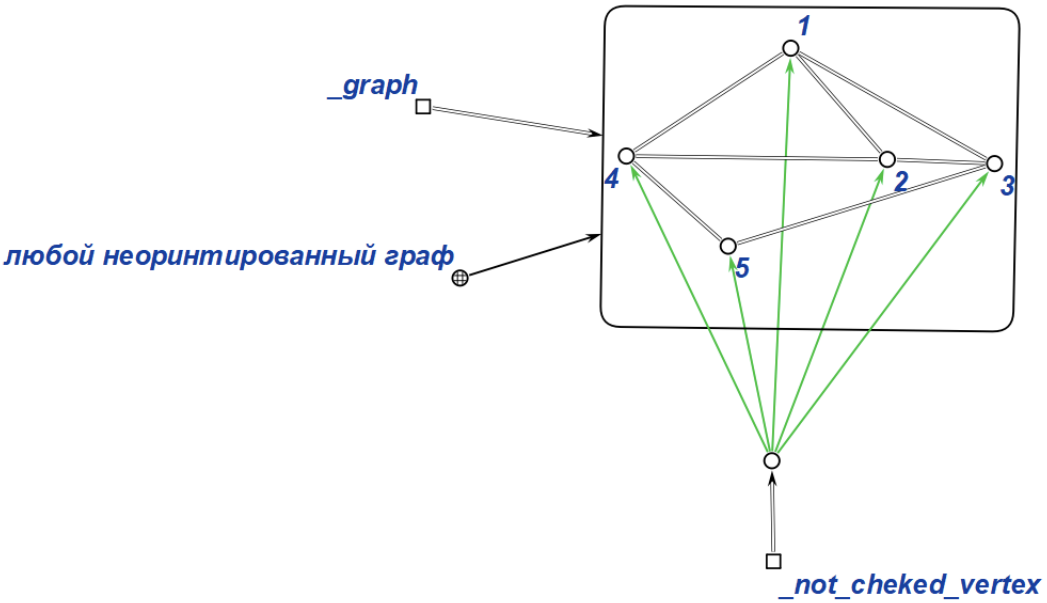

Рис. 14: Действие 1

2. Переменная ***not checked vertexes*** получит в качестве значения множество непроверенных вершин обрабатываемого графа.

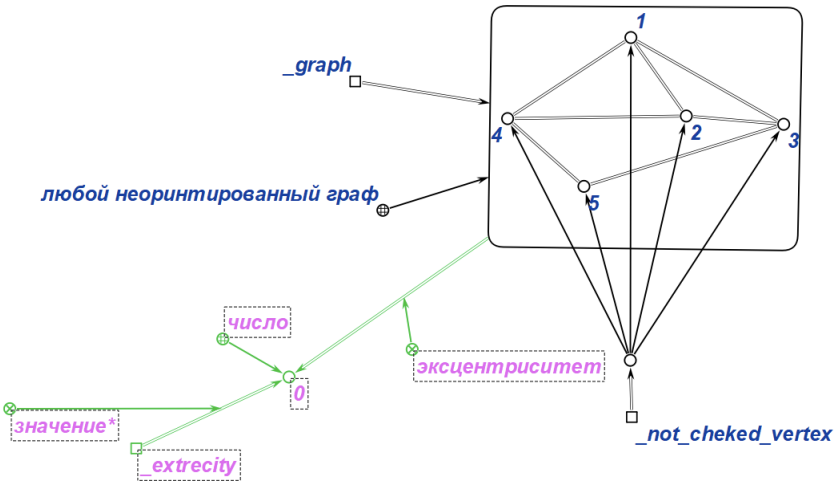

Рис. 15: Действие 2

3. На данном этапе программа создает переменную ***extrecity***, которая будет счетчиком для выбранной нами вершины. Он будет увеличиватся по ходу волны, если будет увеличиваться расстояние между вершинами.

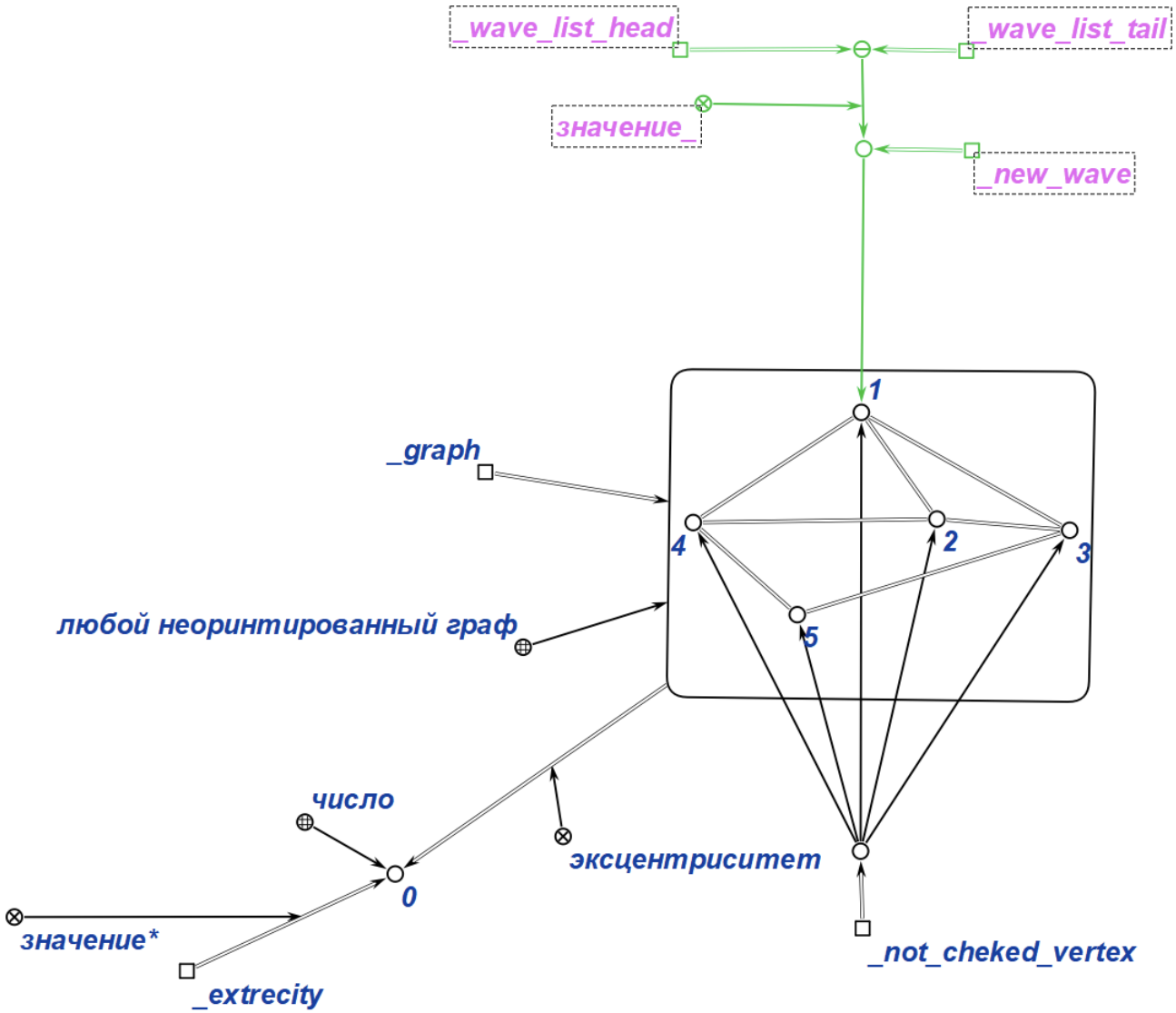

Рис. 16: Действие 3

4. На этом этапе программа создает первую волну из списка волн. Первая волна содержит только начальную вершину 1. Переменная ***new wave*** получает в качестве значения созданную волну, и в будущем будет всегда указывать на вновь созданную волну. Переменная ***waves list head*** указывает на начальный элемент списка волн, а переменная ***waves list tail*** сейчас и в последующих шагах – на концевой элемент списка волн.

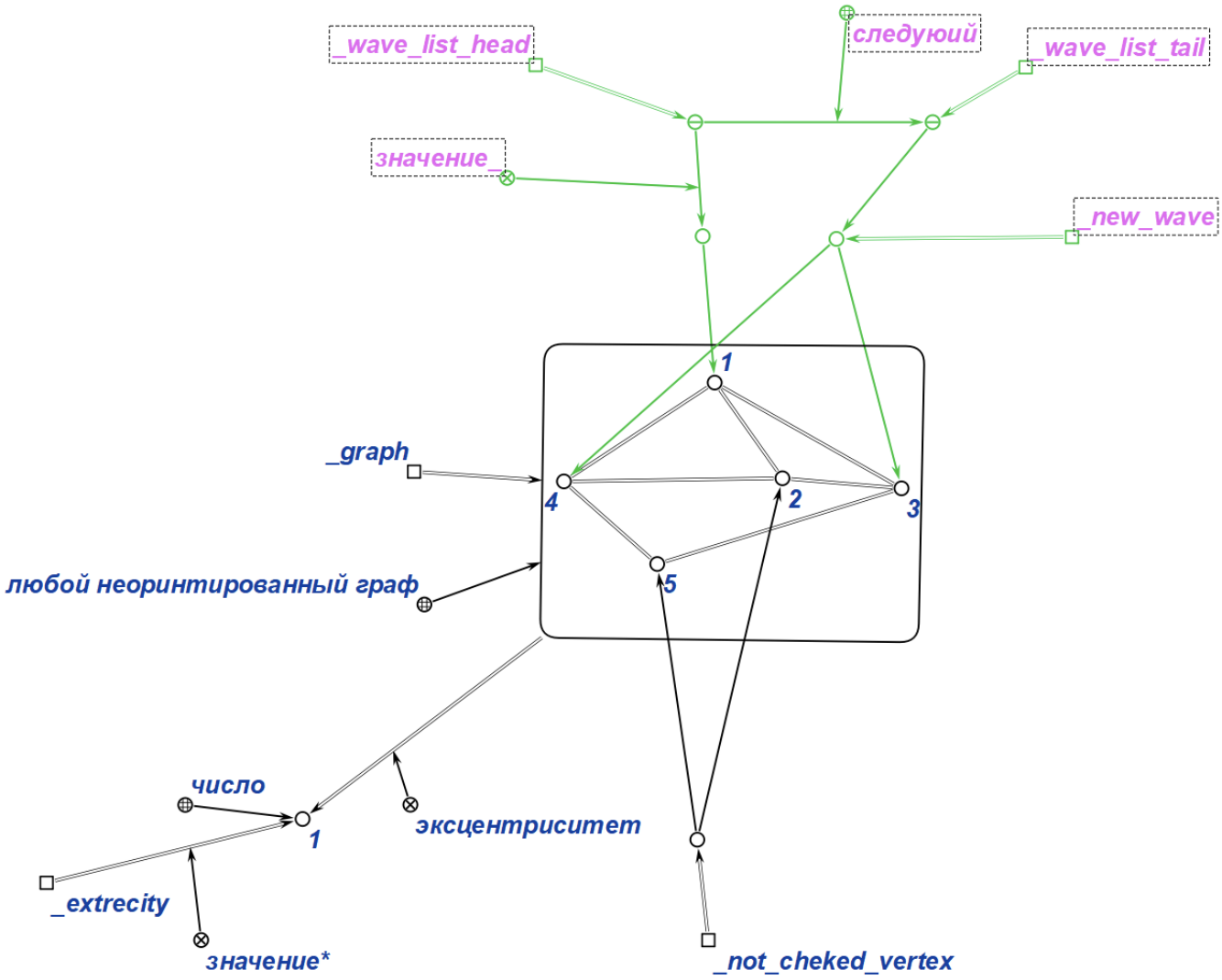

Рис. 17: Действие 4

5. Переменная ***waves list tail*** получает в качестве значения созданный элемент списка, а переменная ***new wave*** – созданную волну. Волна переходит на соседние вершины (3, 4) и эксцентриситет увеличивается на 1.

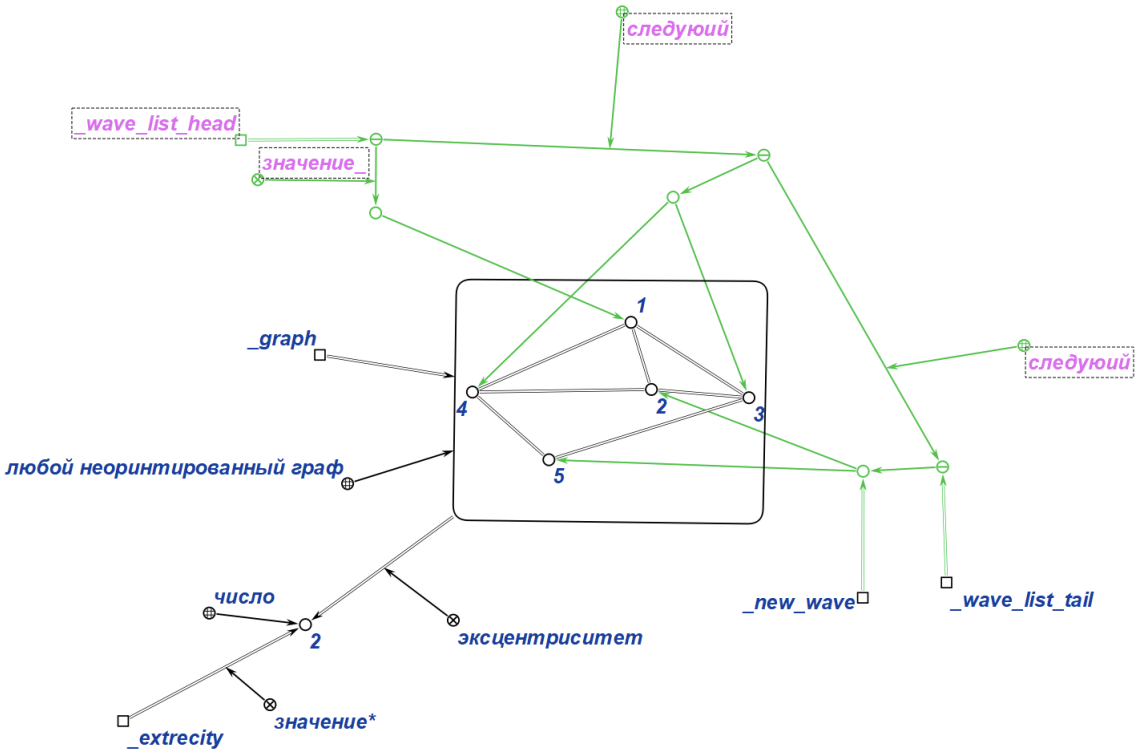

Рис. 18: Действие 5

6. Проходим по следующим соседним вершинам, эксцентриситет увеличивается еще на один (теперь 2), не забываем, что это относительно выбранной нами вершины (1).

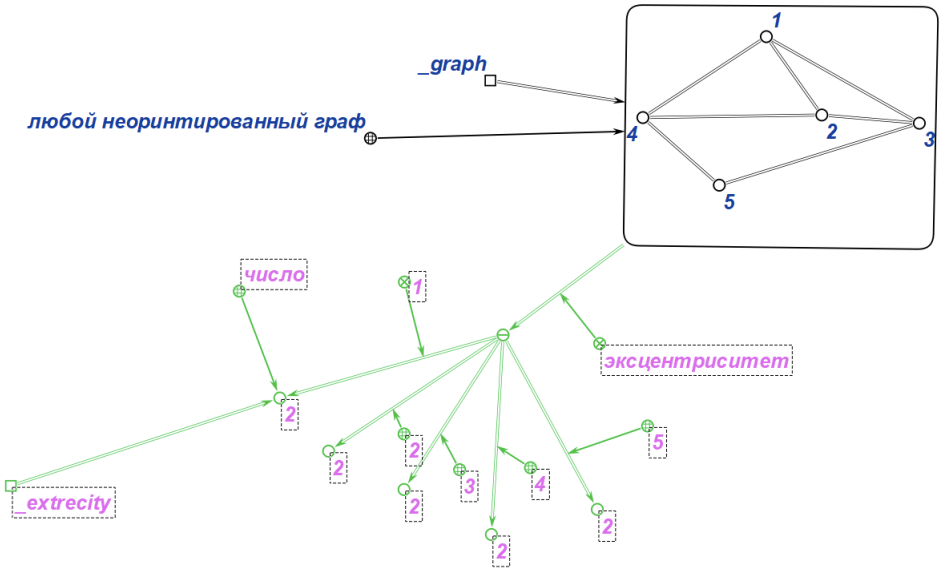

Рис. 19: Действие 6

7. Этот алгоритм прохождения мы повторяем для каждоый вершины. Таким образом, повторив волну для каждой вершины, мы получили значение эксцентриситета для каждой вершины заданного графа.

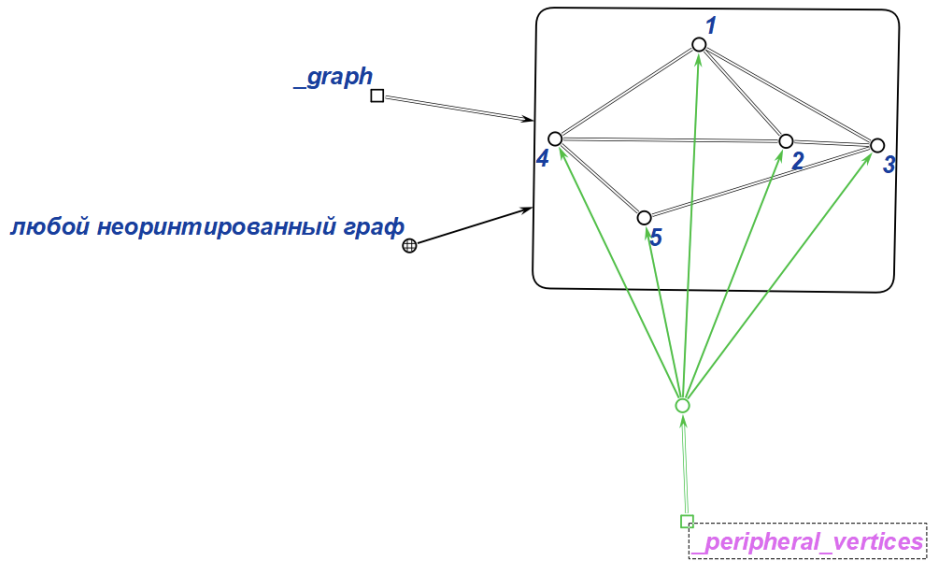

Рис. 20: Действие 6

8. Получив значение эксцентриситета мы можем узнать значение диаметра (наибольший эксцентриситет). После, мы можем узнать какие вершины в этом примере периферийные (в данном - все). Таким образом, мы создаем переменную ***periphera vertices***, которая содержит все периферийные вершины.

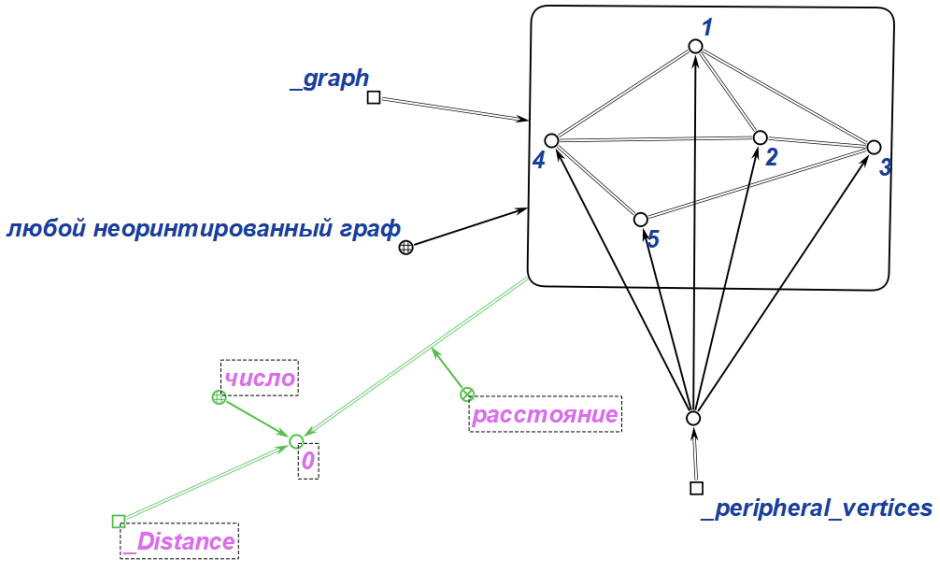

Рис. 21: Действие 7

9. На данном этапе создаем переменную ***Distance***, которая будет счетчиком для выбранной нами вершины. Он будет увеличиватся по ходу волны, если будет увеличиваться расстояние между вершинами. И тогда мы сможем узнать расстоянииие от выбранной периферийной вершины до остальных.

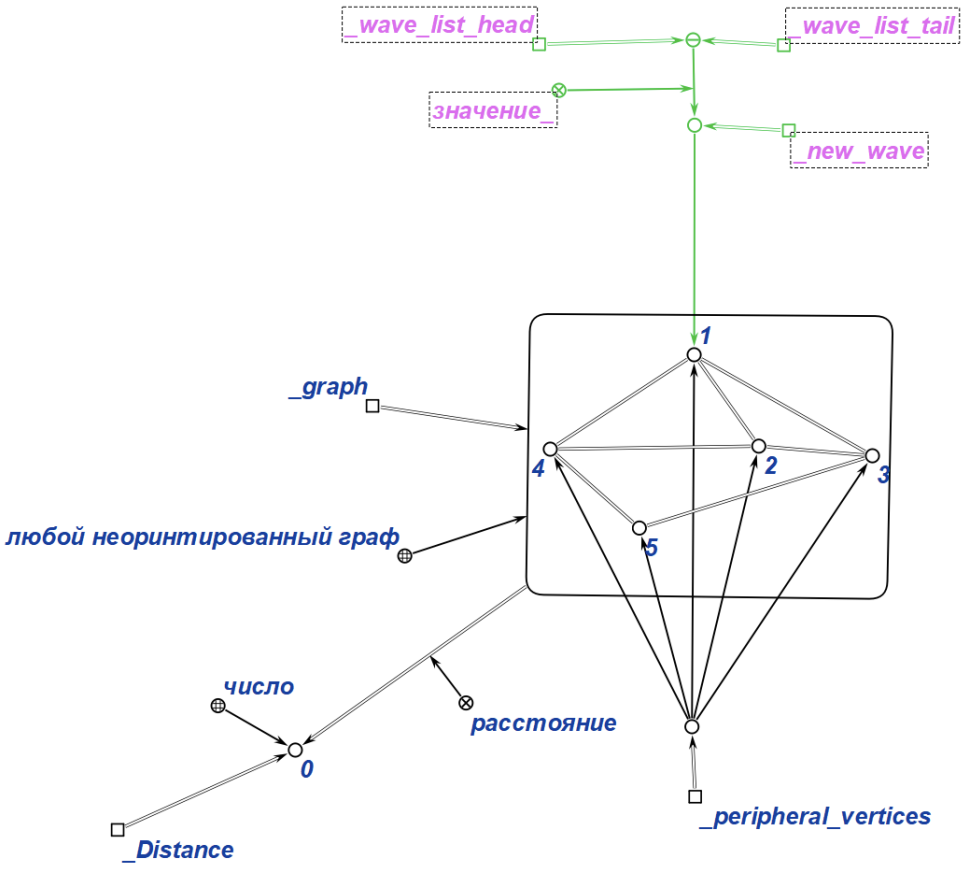

Рис. 22: Действие 8

10. Мы запускаем такую же волну, как и та, что была в начале описания. Принцип работы будет одиновый, только "счетчик"другой, как я говорила ранее, теперь мы ищем расстояние.

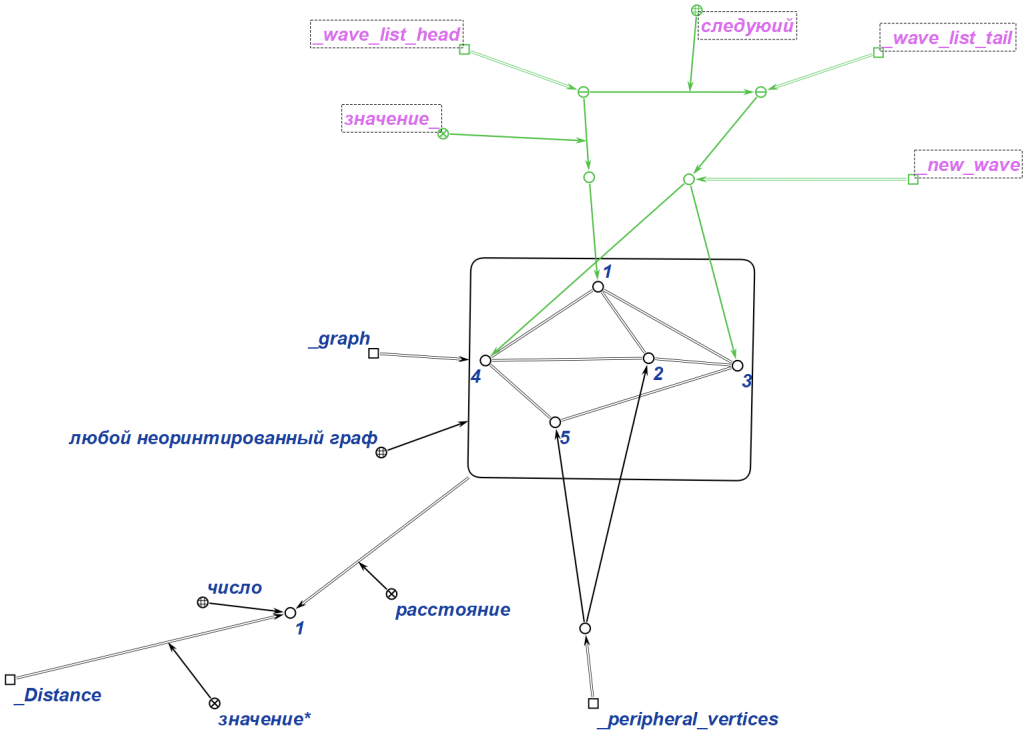

Рис. 23: Действие 9

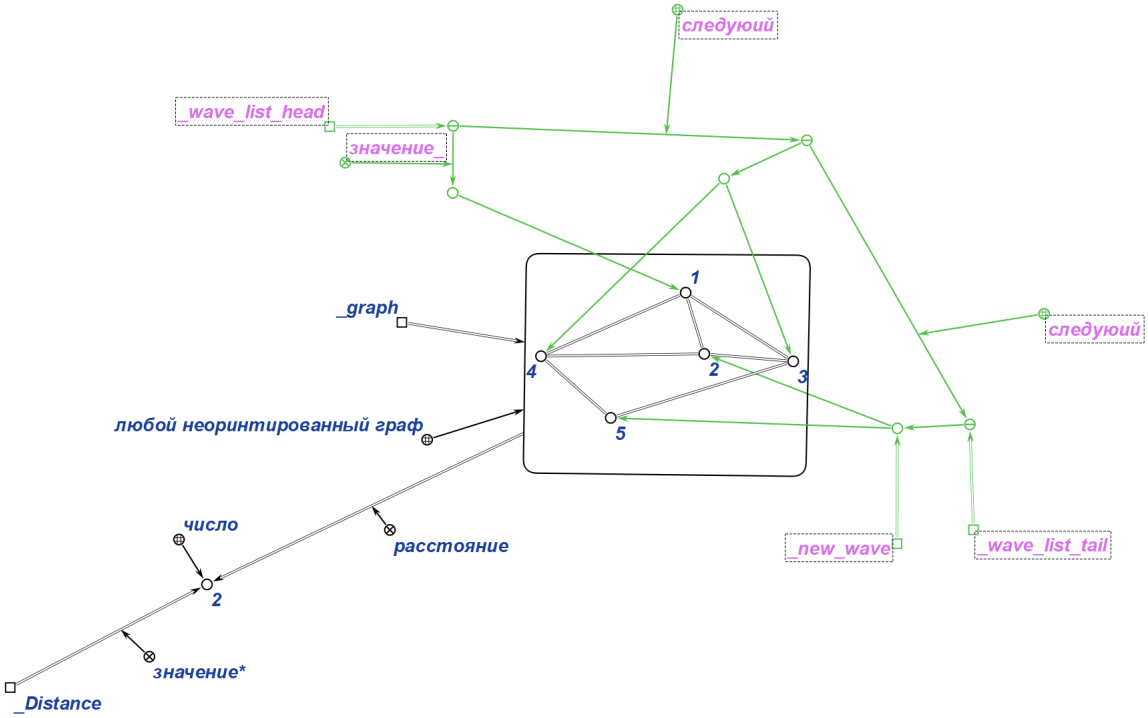

Рис. 24: Действие 10

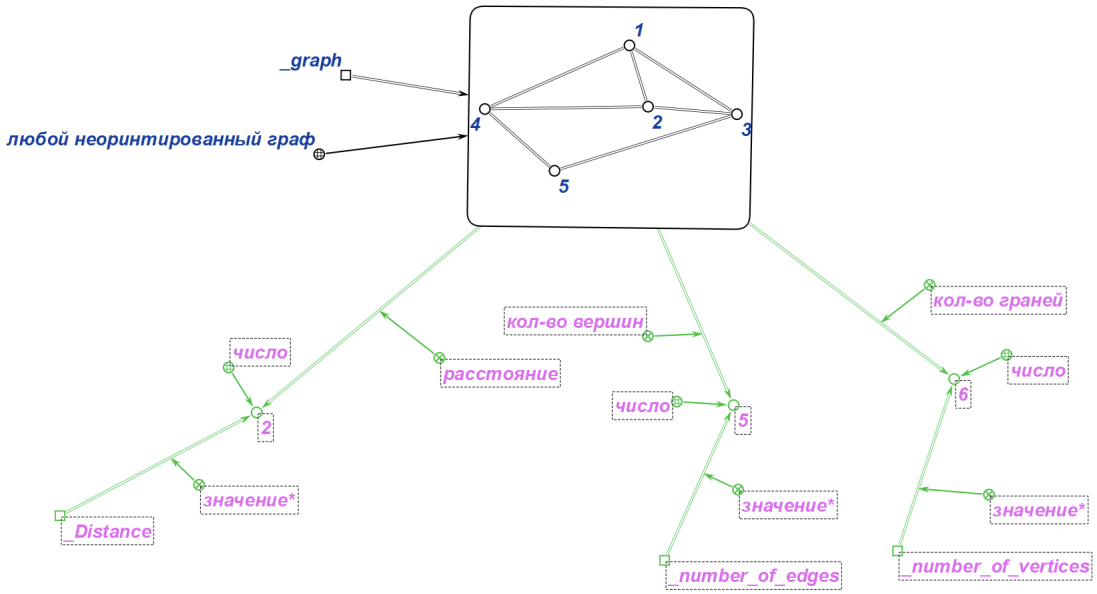

Рис. 25: Действие 11

11. В результате обхода графа этой волной мы получили значение расстоянния выбранной нами вершины. Это действие повторяется для всех периферийных вершин. Так же, обойдя весь граф, в переменных ***number of edges***, ***number of vertices*** мы храним кол-во вершин и кол-во граней (далее понадобится).

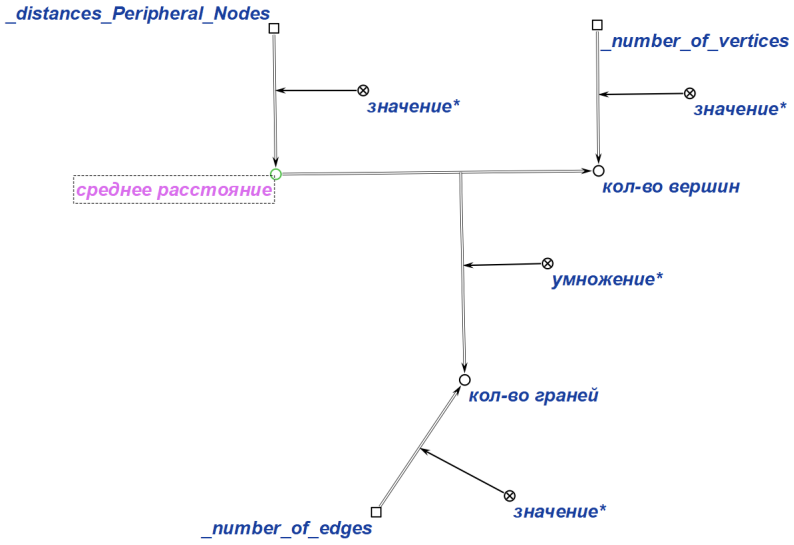

Рис. 26: Действие 12

12. Зная расстояния между всеми периферийными вершинами мы можем расчитать значение среднего расстояние (оно будет храниться в переменной ***distances Peripheral Nodes***).

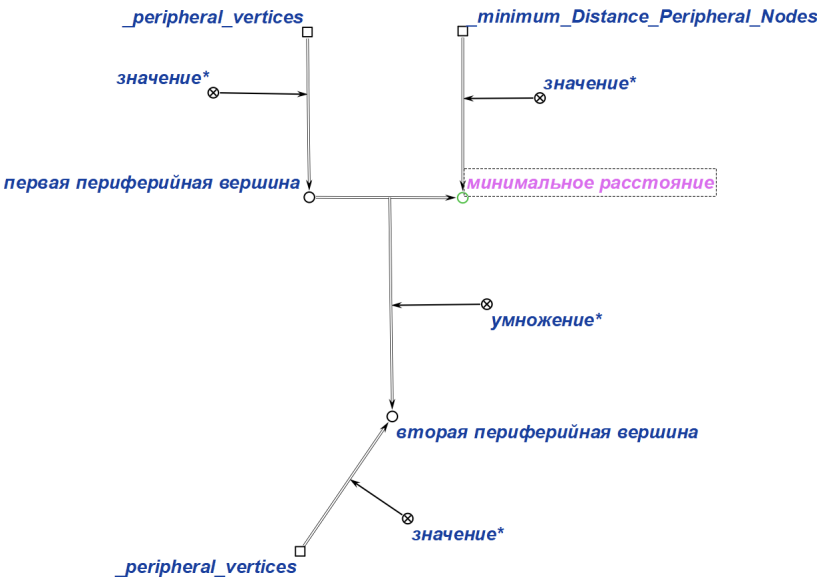

Рис. 27: Действие 13

13. Зная кол-во вершин и кол-во граней (естественно между периферийными вершинами) мы рассчитаем минимальное расстояние(оно будет храниться в переменной ***minimum Distance Peripheral Nodes***).

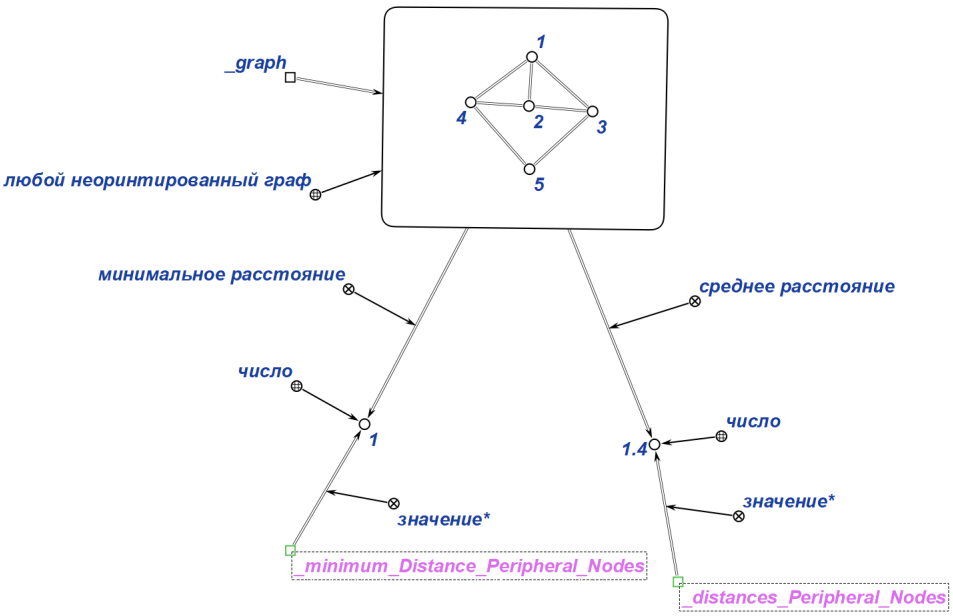

Рис. 28: Выход теста 5

14. Таким образом, на выходе, мы получим две переменные (***minimum Distance Peripheral Nodes***,***distances Peripheral Nodes***), которые будут содержать в себе то, что мы искали.

5 Заключение

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