

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 диаметр взвешенного неориентированного графа.

Выполнил:

Р. Э. Халилов

Студент группы
321702

Проверил:

Н. В. Малиновская

Минск 2024

Содержание

1	Введение	2
2	Список понятий	2
3	Тестовые примеры	6
3.1	Тест 1	6
3.2	Тест 2	8
3.3	Тест 3	9
3.4	Тест 4	10
4	Пример работы алгоритма в семантической памяти	11
4.1	Краткое описание:	11
4.2	Демонстрация на тесте 5:	11
5	Заключение	17

1 Введение

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

Задача: Найти диаметр взвешенного 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 существен

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

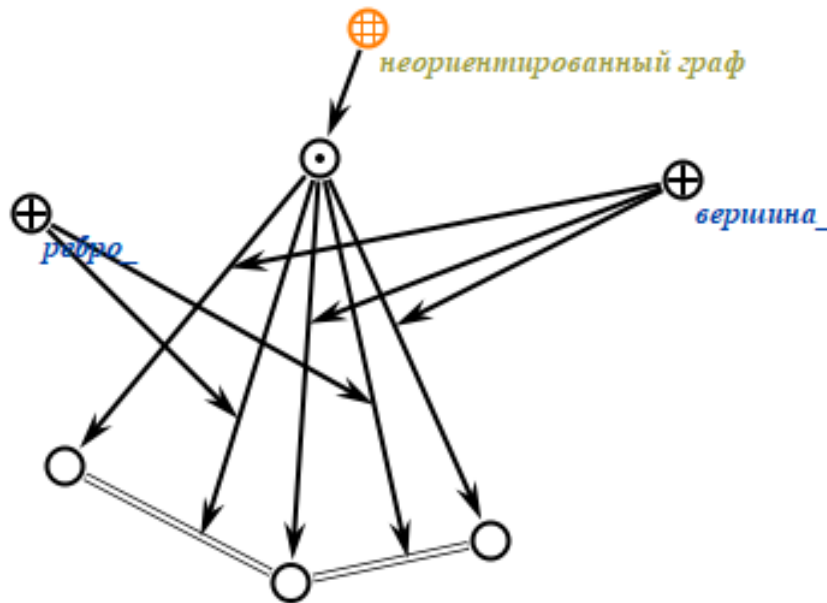


Рис. 1: Абсолютное понятие неориентированного графа

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

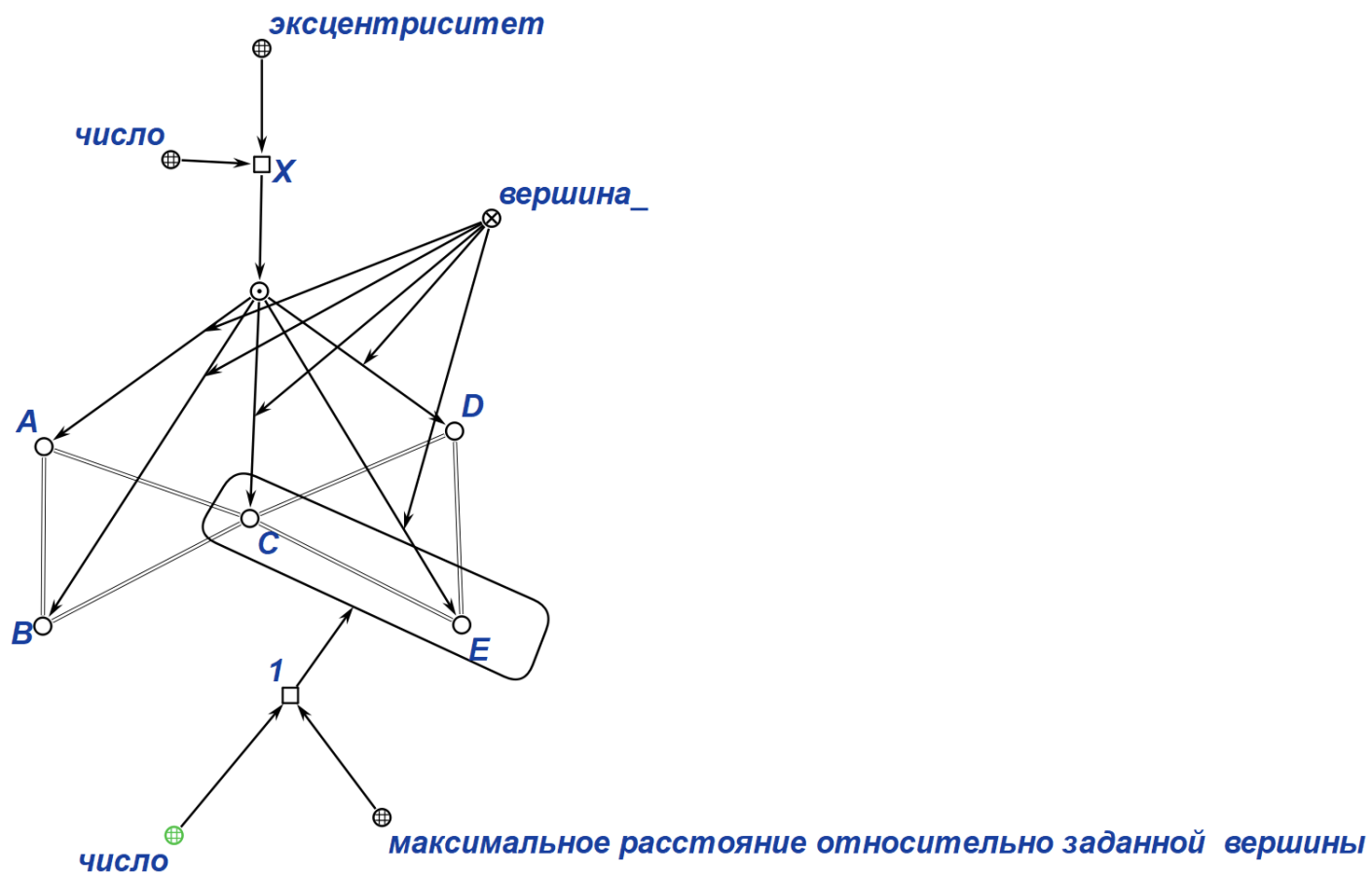


Рис. 2: Абсолютное понятие эксцентриситета

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

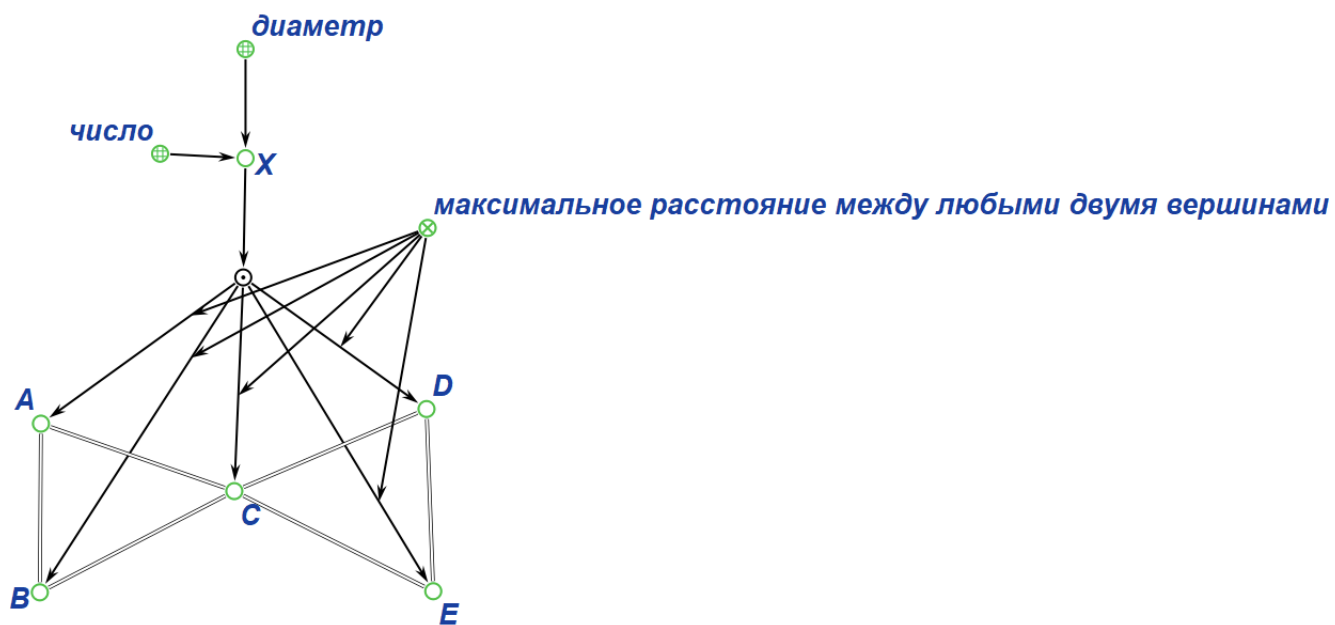


Рис. 3: Абсолютное понятие диаметра

4. **Взвешенный граф** (абсолютное понятие)-граф, каждому ребру которого поставлено в соответствие некое значение (вес ребра)
- (a) Вершина (относительное понятие, ролевое отношение);
 - (b) Связка (относительное понятие, ролевое отношение);
 - (c) Вес (относительное понятие, отношение);

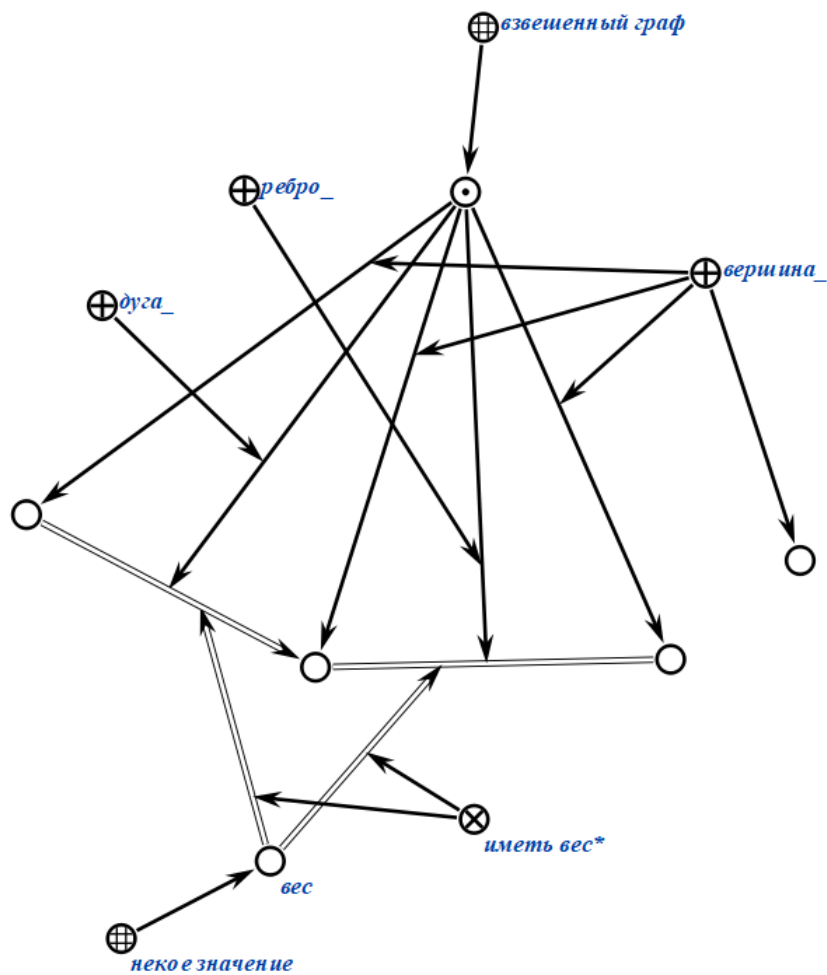


Рис. 4: Абсолютное понятие взвешенного графа

3 Тестовые примеры

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

3.1 Тест 1

Вход: Необходимо найти диаметр графа.

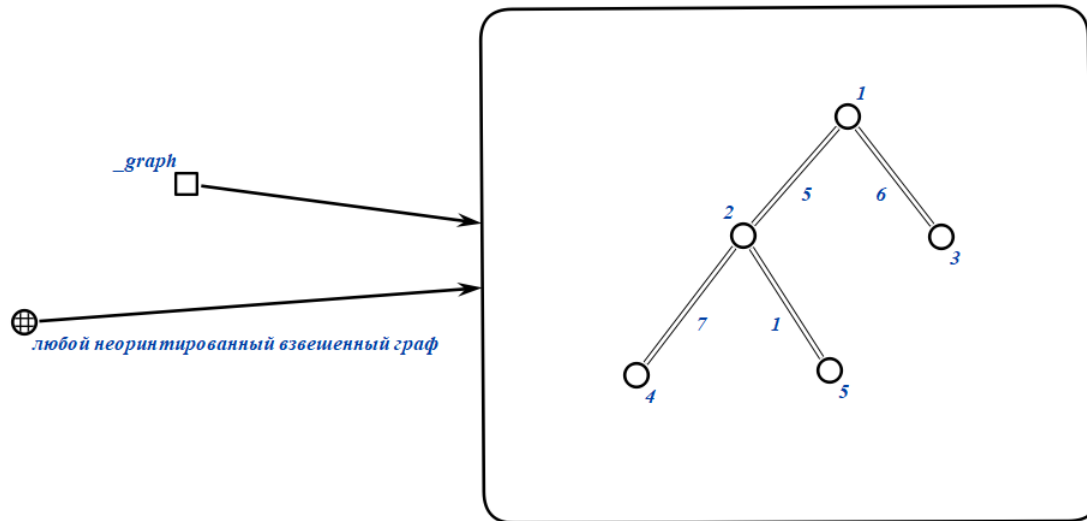


Рис. 5: Вход теста 1

Выход: Будет найдено значение 18, равное димаетру графа.

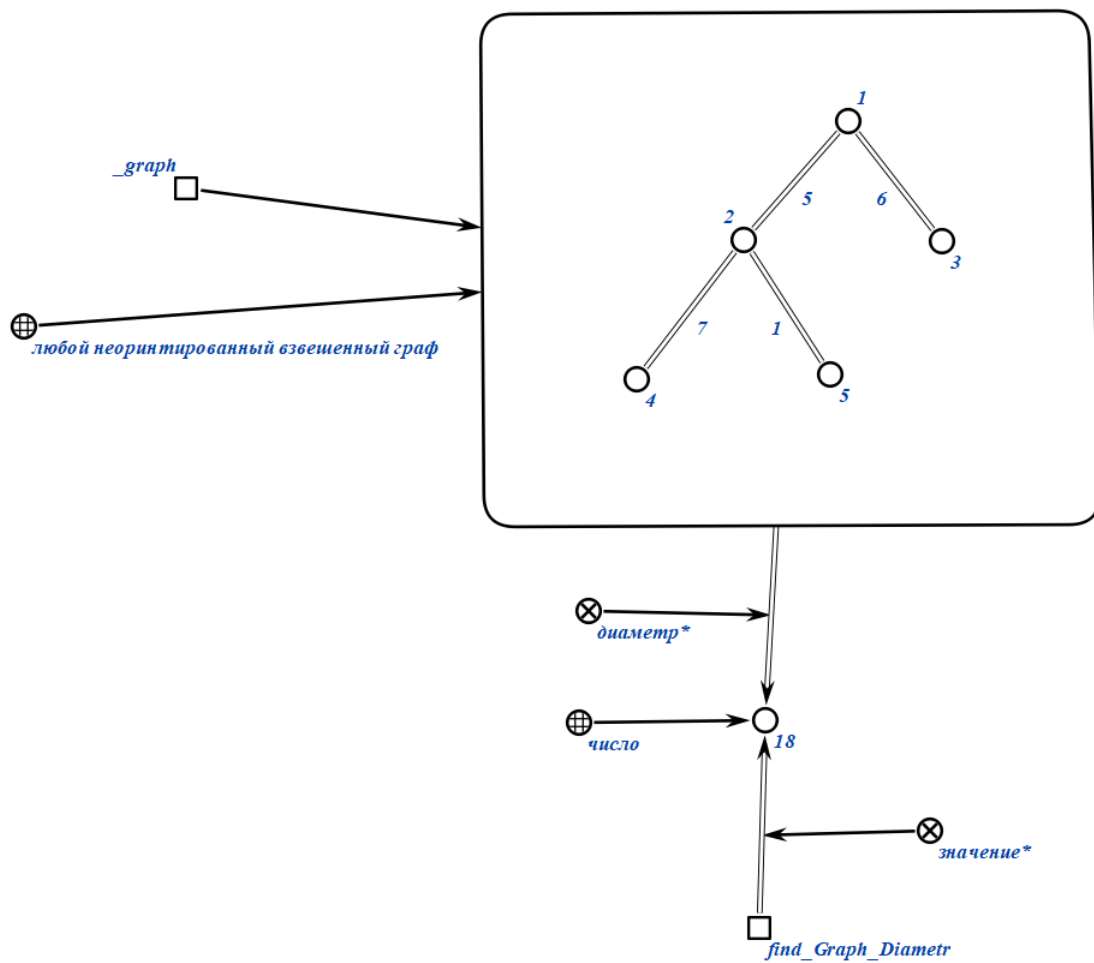


Рис. 6: Выход теста 1

3.2 Тест 2

Вход: Необходимо найти диаметр графа.

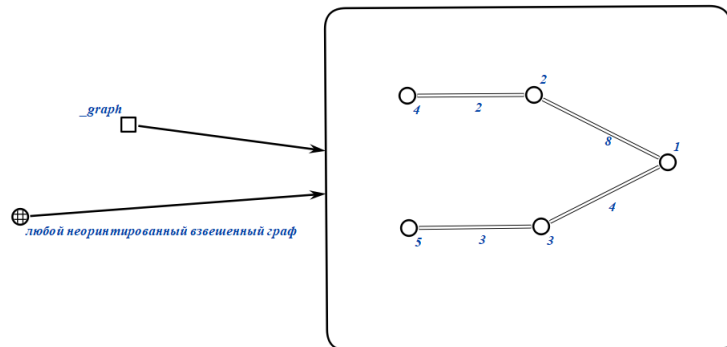


Рис. 7: Вход теста 2

Выход: Будет найдено значение 17, равное диаметру графа.

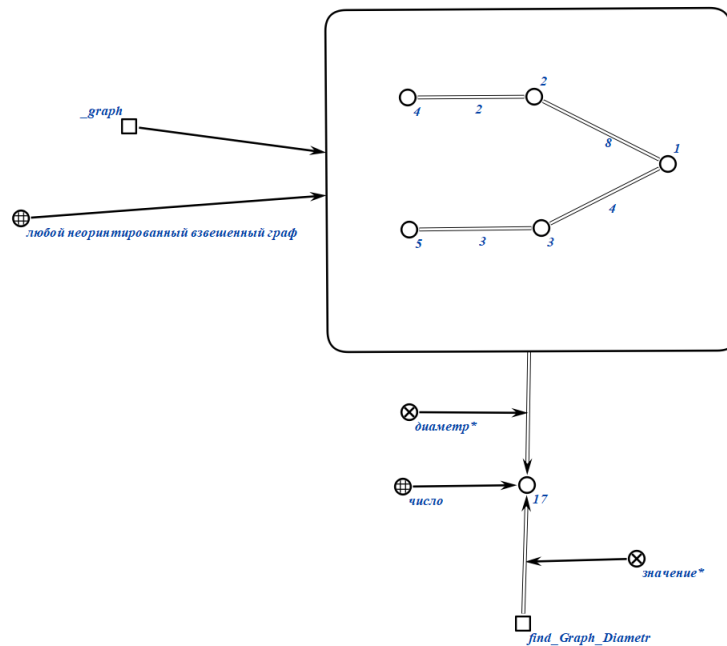


Рис. 8: Выход теста 2

3.3 Тест 3

Вход: Необходимо найти диаметр графа.

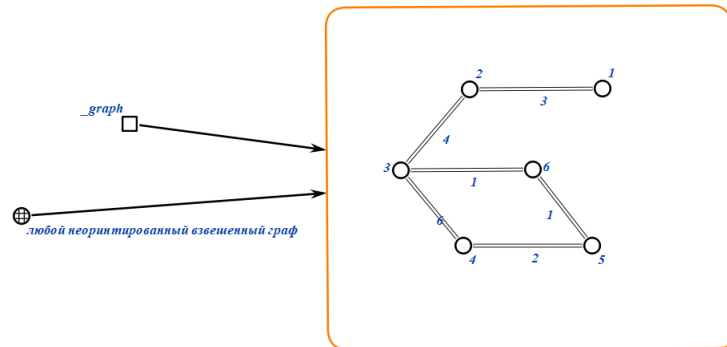


Рис. 9: Вход теста 3

Выход: Будет найдено значение 11, равное диаметру графа.

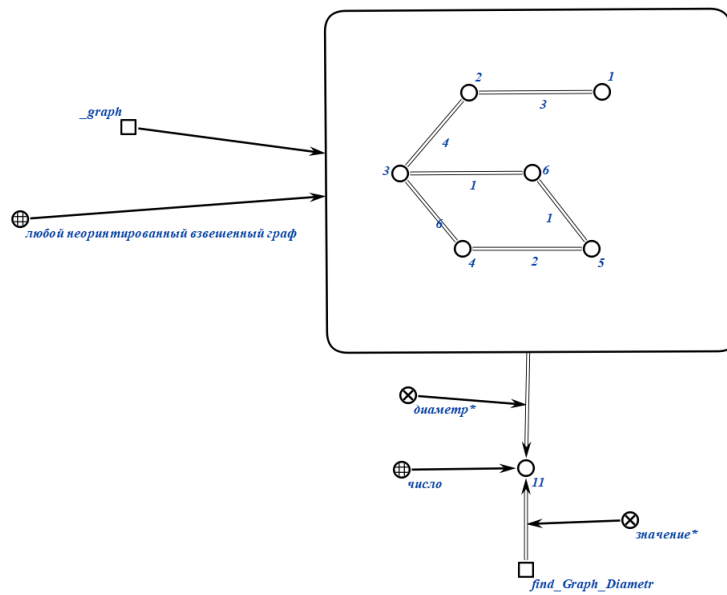


Рис. 10: Выход теста 3

3.4 Тест 4

Вход: Необходимо найти диаметр графа.

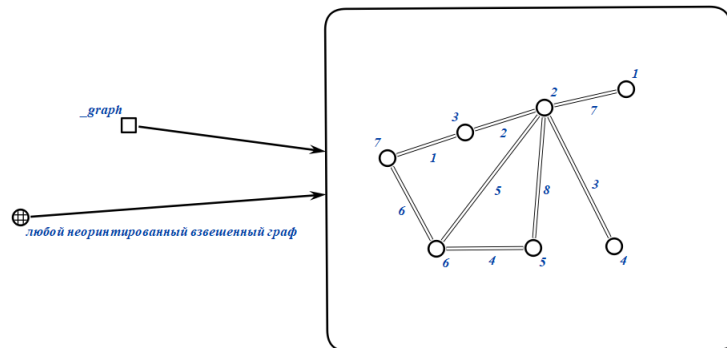


Рис. 11: Вход теста 4

Выход: Будет найдено значение 15, равное диаметру графа.

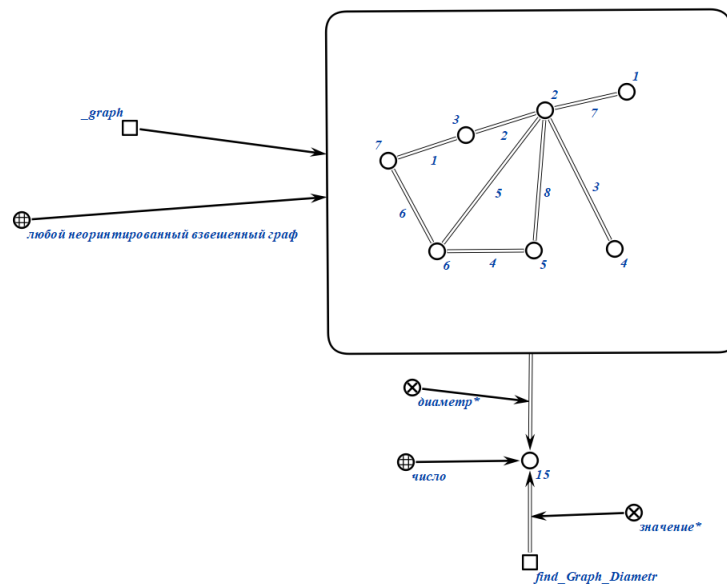


Рис. 12: Выход теста 4

4 Пример работы алгоритма в семантической памяти

4.1 Краткое описание:

- (a) первый элемент первого уровня содержит список
- (b) Добавить все вершины графа, кроме вершины, которую мы выбрали, во множество непосещенных вершин.
- (c) Создаем переменную (счетчик), в которой будет храниться эксцентриситет данной вершины.
- (d) Создать новую волну и добавить в нее вершину, которая еще не посещена.
- (e) При запуске новой волны счетчик будет увеличиваться на 1, если эксцентриситет увеличивается (если нет - счетчик не изменяется) .
- (f) Начальная волна - это новая волна. Новой волной будем называть последнюю созданную волну.
- (g) Сформировать следующую волну для новой волны. В нее попадут 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.
- (h) Если новая волна пуста, то мы обошли весь граф. И получили значение эксцентриситетов всех вершин.
- (i) Сравниваем эксцентриситеты и находим диаметр графа.
- (j) Выводим полученный результат.

4.2 Демонстрация на тесте 5:

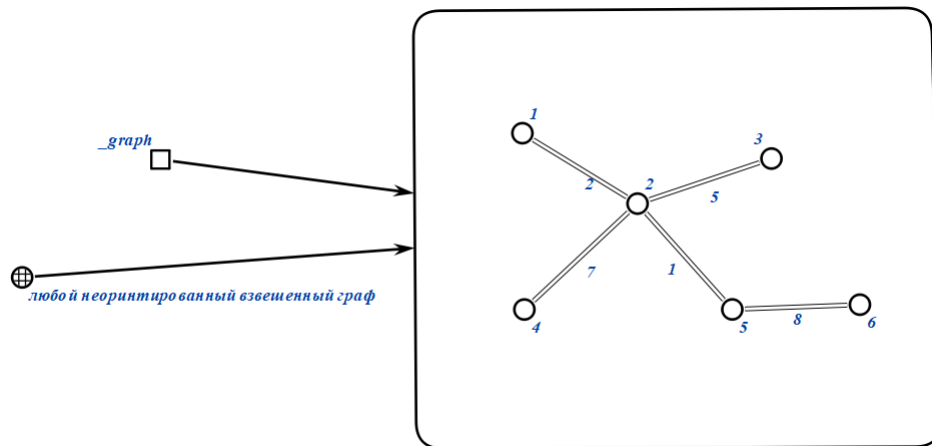


Рис. 13: Вход теста 5

- (a) **graph** получит в качестве значения sc-узел неориентированного графа;

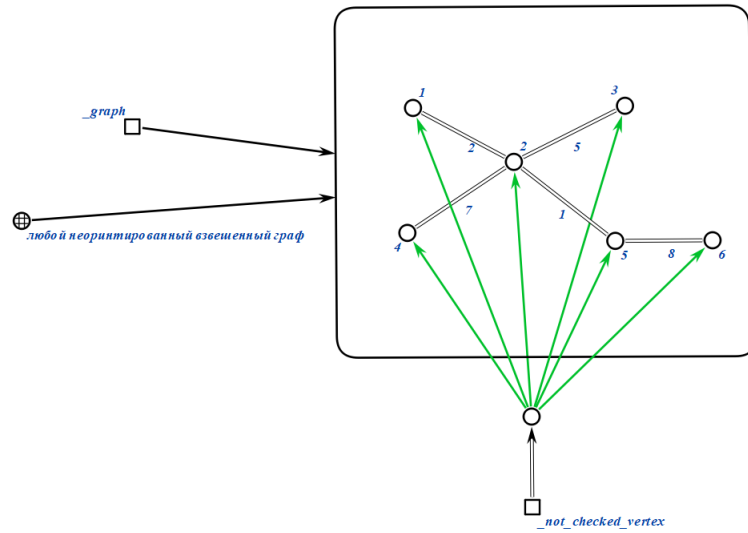


Рис. 14: Действие 1

- (b) Переменная *not checked vertices* получит в качестве значения множество непроверенных вершин обрабатываемого графа.

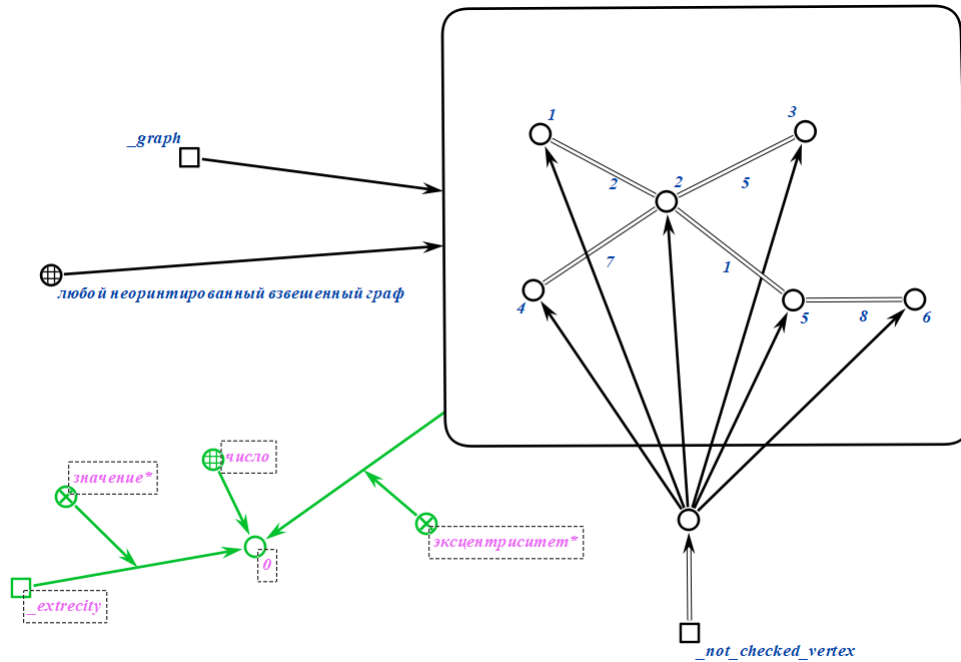


Рис. 15: Действие 2

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

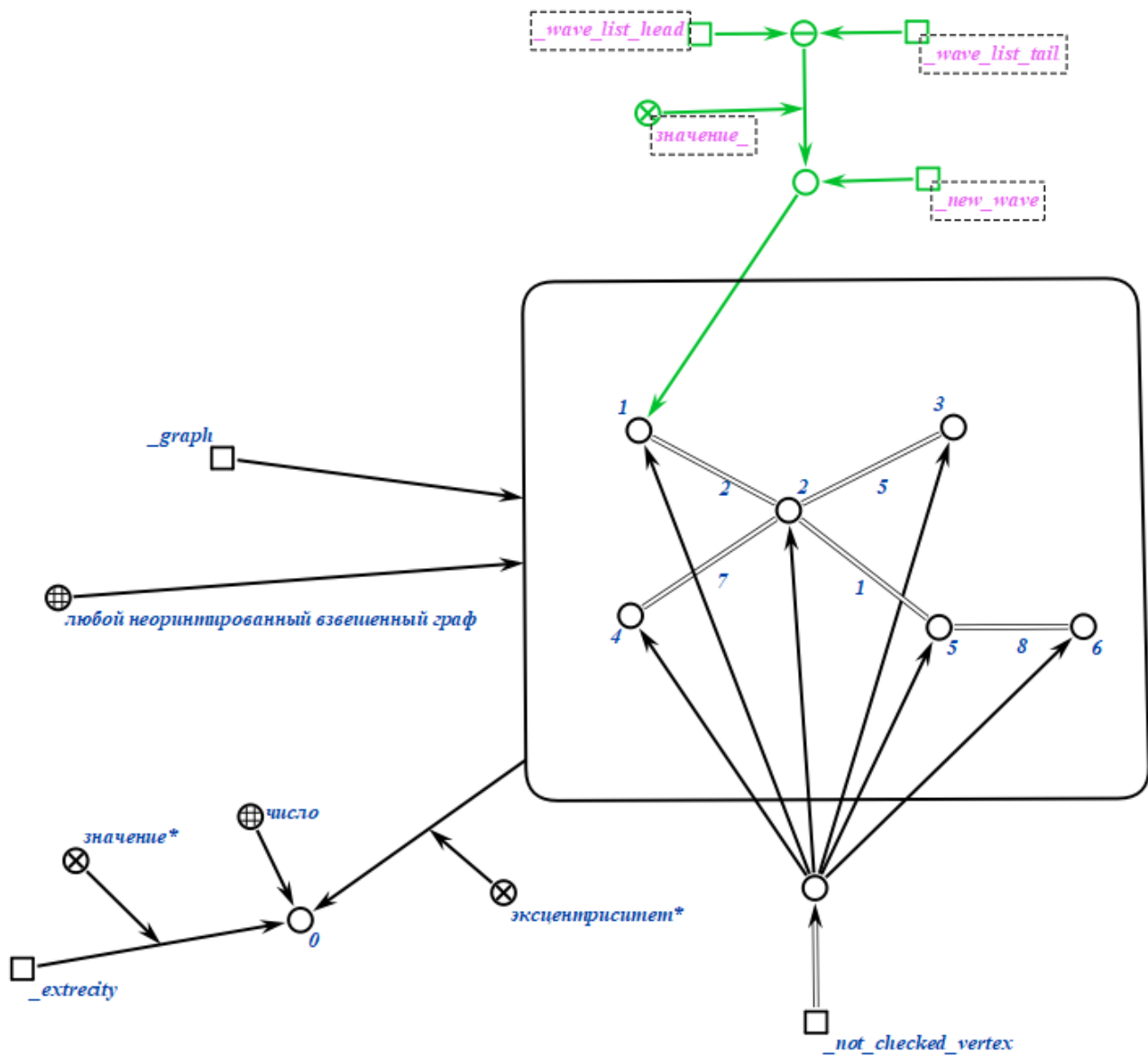


Рис. 16: Действие 3

- (d) 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.

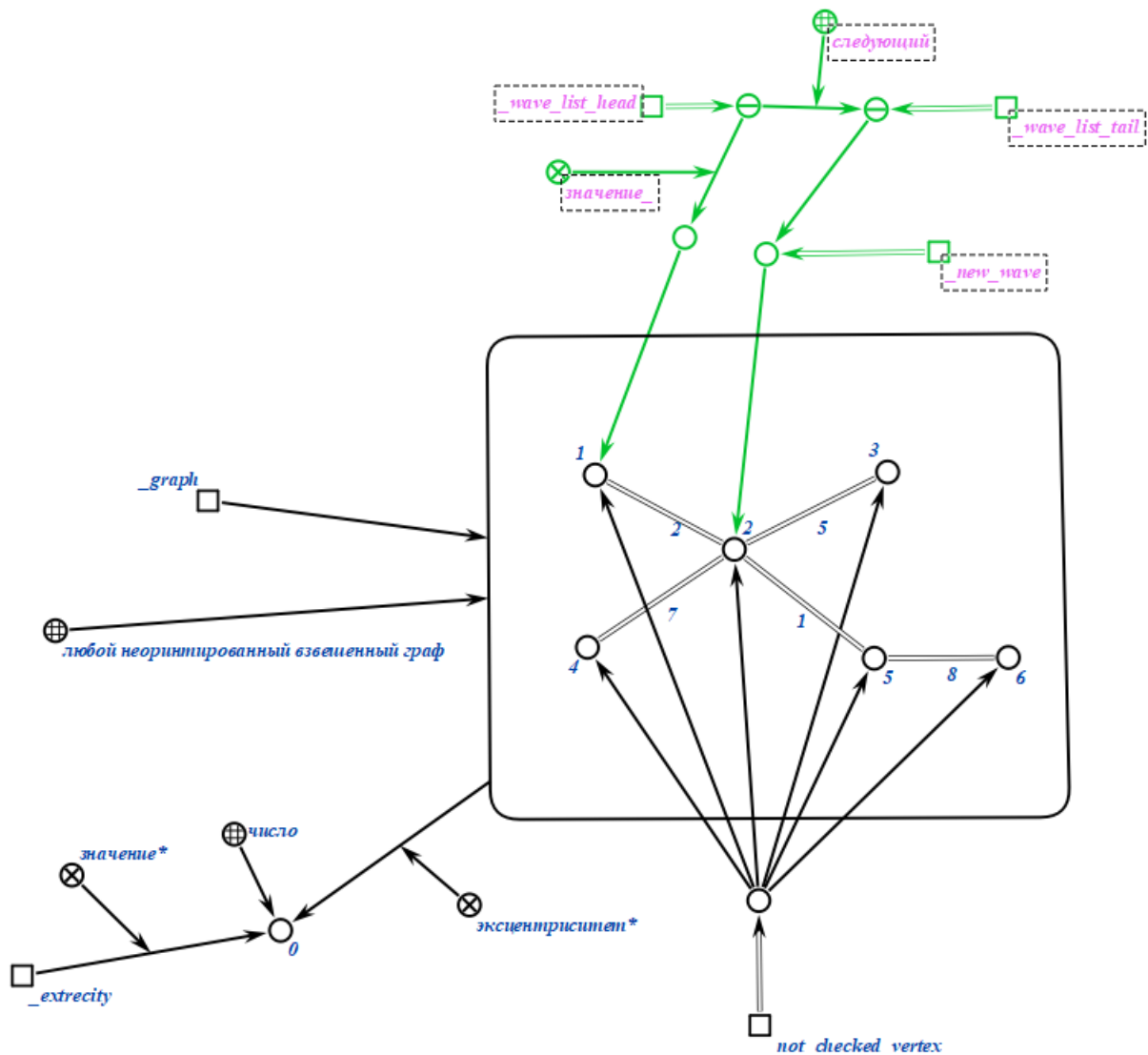


Рис. 17: Действие 4

- (е) Переменная `waves list tail` получает в качестве значения созданный элемент списка, а переменная `new wave` – созданную волну. Волна переходит на соседнюю вершину (2) и эксцентриситет увеличивается на 2.

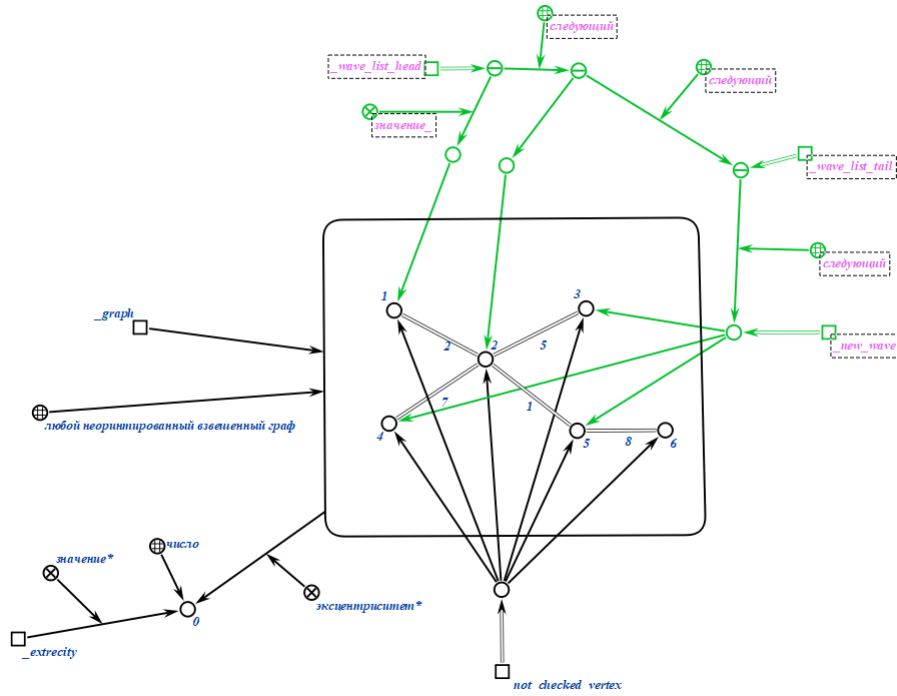


Рис. 18: Действие 5

- (f) Проходим по следующим соседним вершинам, увеличивая эксцентриситет на большее из найденных значений веса (теперь 9). Не забываем, что это относительно выбранной вершины (1). Продолжив алгоритм после вершины (5), находим, что эксцентриситет относительно вершины (1) равен 11.

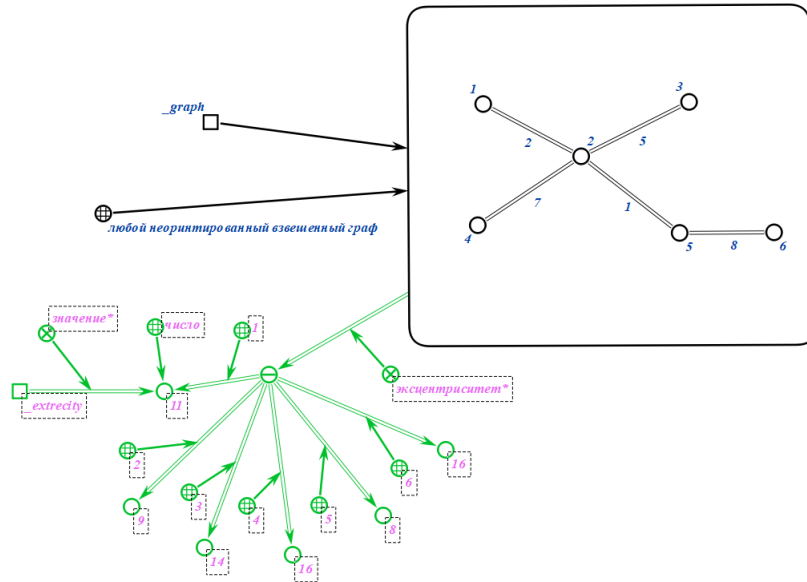


Рис. 19: Действие 6

- (g) Этот алгоритм прохождения мы повторяем для каждой вершины. Таким образом, повторив волну для каждой вершины, мы получили значение эксцентриситета для каждой вершины заданного графа.

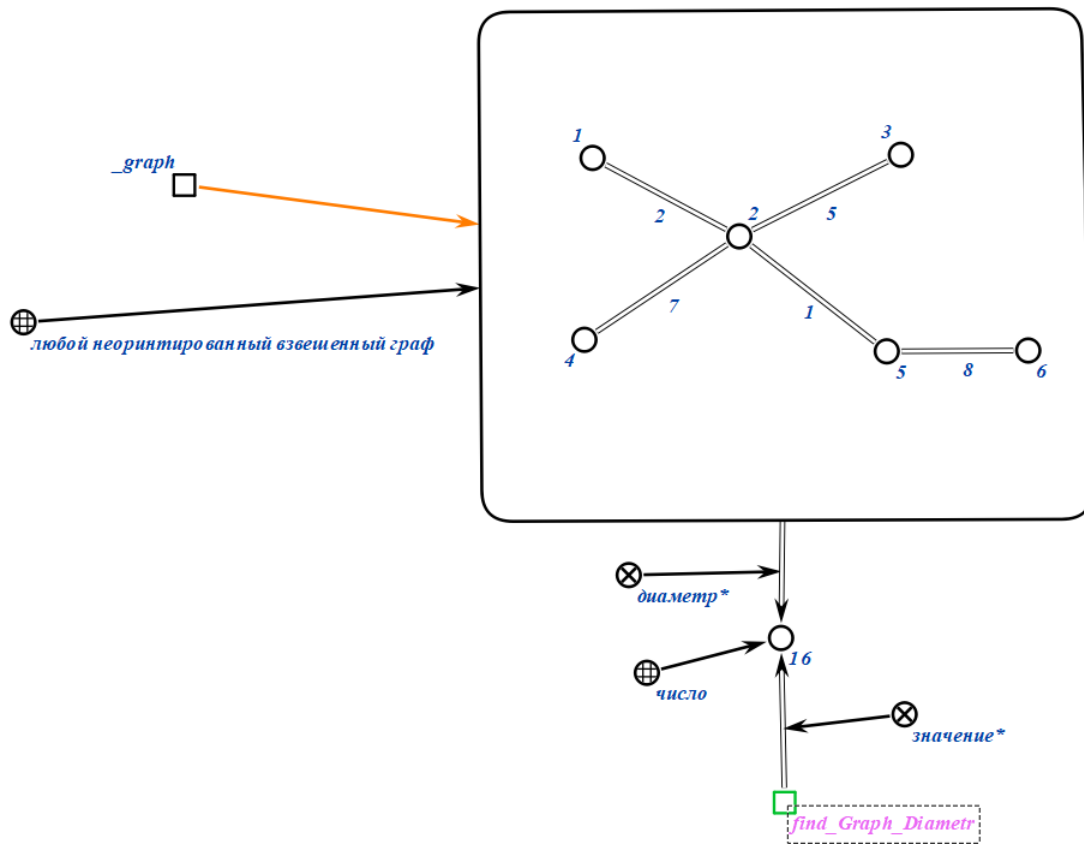


Рис. 20: Действие 7

- (h) Получив значение эксцентриситета мы можем узнать значение диаметра (наибольший эксцентриситет). Таким образом, на выходе мы получим переменную ***Find Graph Diameter***, которая будет содержать в себе значение диаметра приведенного в пример графа.

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 взвешенном связном графе.