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

Решение теоретико-графовой задачи. Задача нахождения радиуса взвешенного неориентированного графа

Выполнила: П. В. Пучинская

Студент группы
321702
Проверил: Н. В. Малиновская

Минск 2024

1 Введение

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

Задача: Найти радиус взвешенного неориентированного графа.

2 Список понятий

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

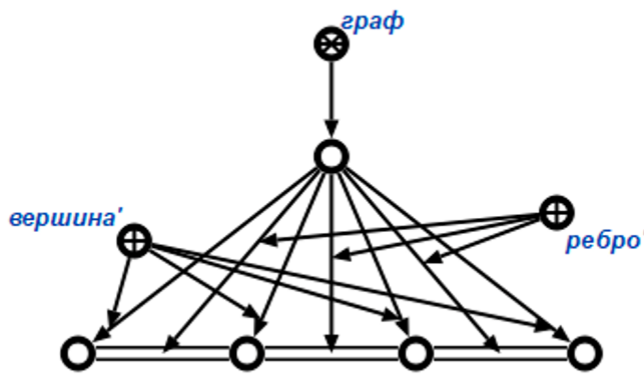

Рис. 1: Граф

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

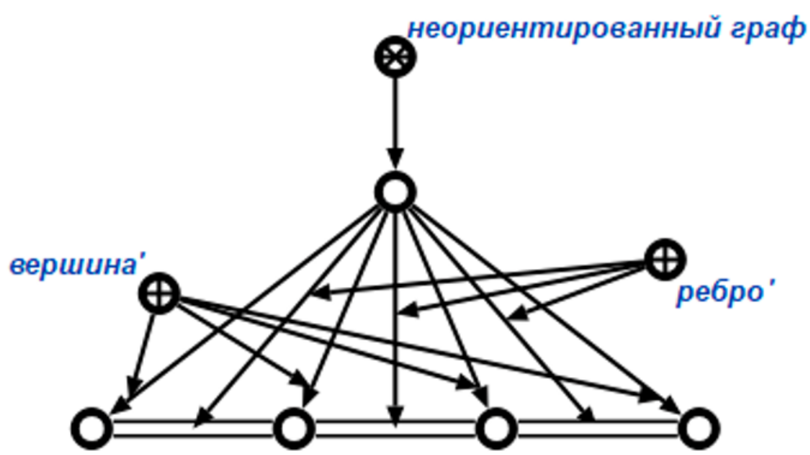

Рис. 2: Неориентированный граф

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

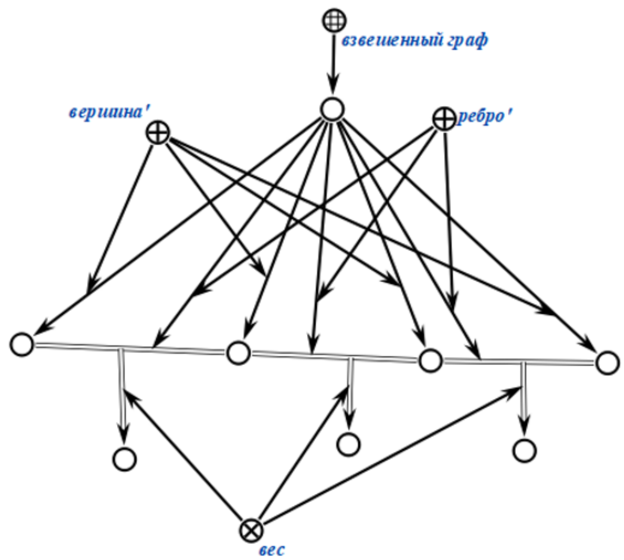

Рис. 3: Взвешенный граф

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

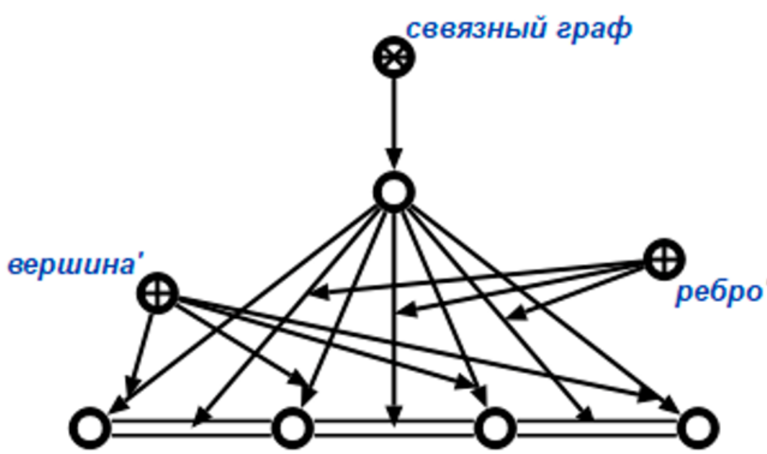

Рис. 4: Связный граф

5. ***Эксцентриситет вершины*** (Рис.5) (абсолютное понятие) – расстояние до самой дальней вершины графа.

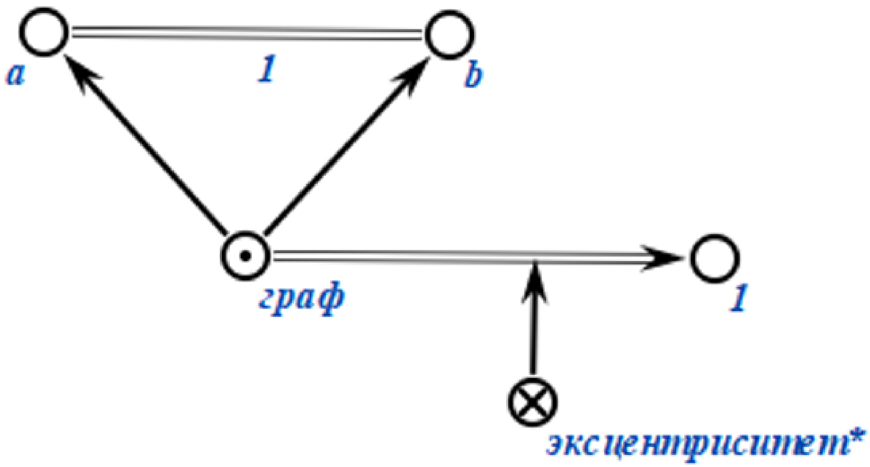

Рис. 5: Эксцентриситет

6. ***Радиус графа*** (Рис.6) (абсолютное понятие) – минимальный эксцентриситет среди всех вершин графа.

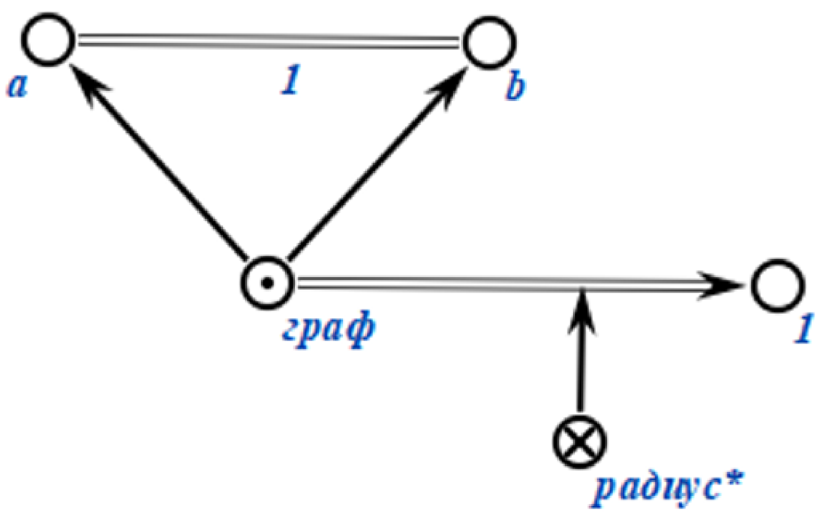

Рис. 6: Радиус

3 Тестовые примеры

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

3.1 Тест 1 (Рис.7, Рис.8)

Вход: Необходимо найти радиус взвешенного неориентированного графа.

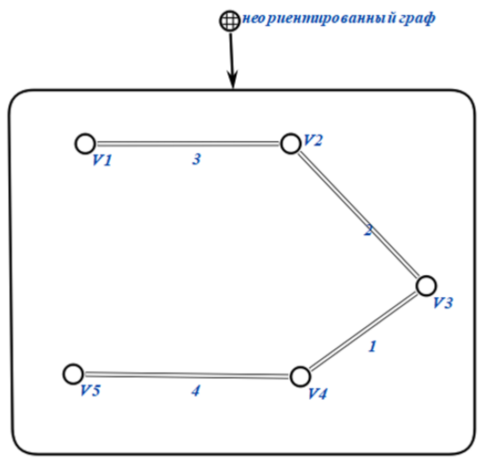

Рис. 7: Вход теста 1

Выход: Радиус графа равен 5.

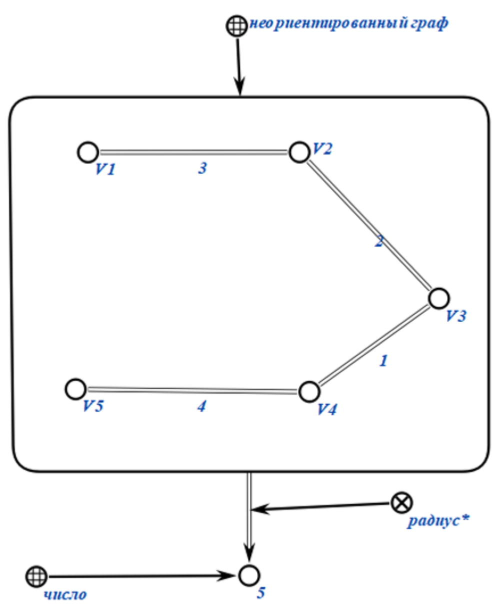

Рис. 8: Выход теста 1

3.2 Тест 2 (Рис.9, Рис.10)

Вход: Необходимо найти радиус взвешенного неориентированного графа.

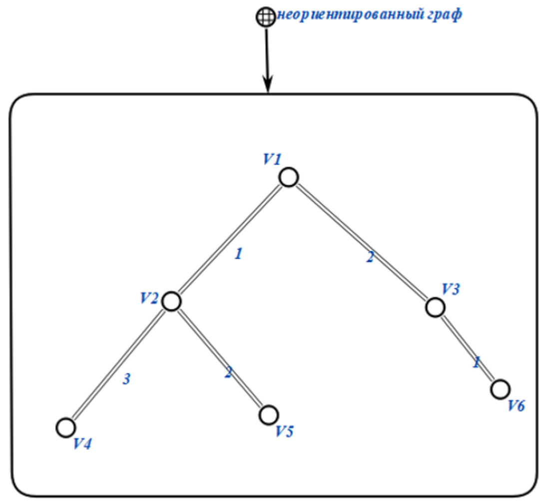

Рис. 9: Вход теста 2

Выход: Радиус графа равен 4.

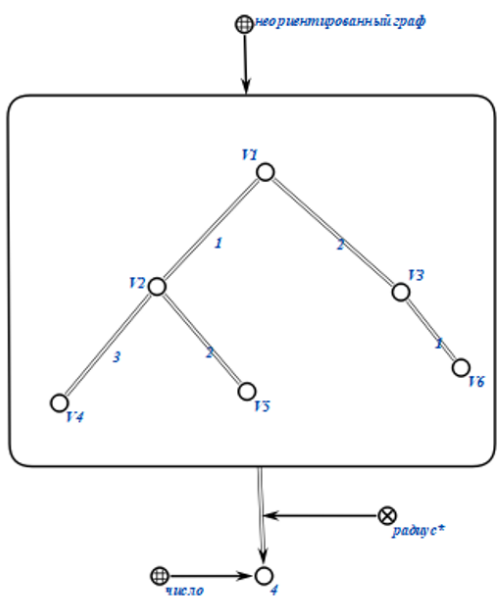

Рис. 10: Выход теста 2

3.3 Тест 3 (Рис.11, Рис.12)

Вход: Необходимо найти радиус взвешенного неориентированного графа.

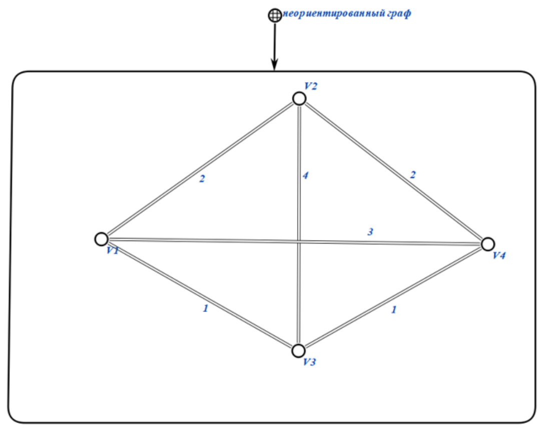

Рис. 11: Вход теста 3

Выход: Радиус графа равен 2.

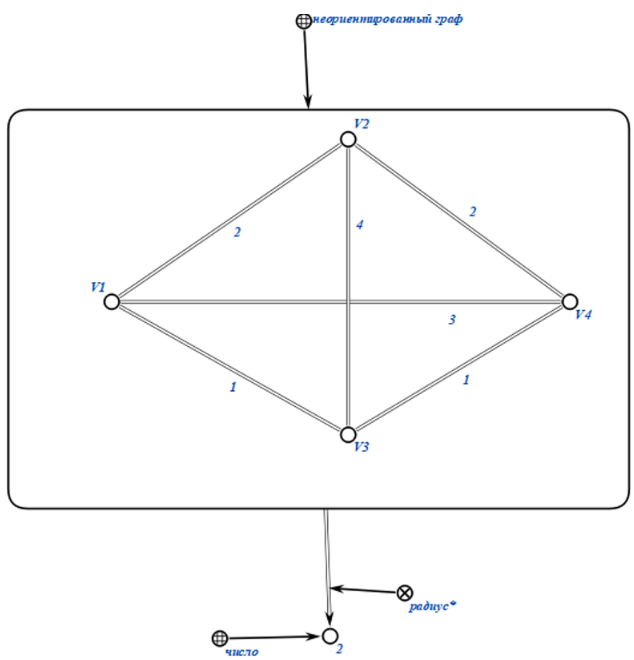

Рис. 12: Выход теста 3

3.4 Тест 4 (Рис.13, Рис.14)

Вход: Необходимо найти радиус взвешенного неориентированного графа.

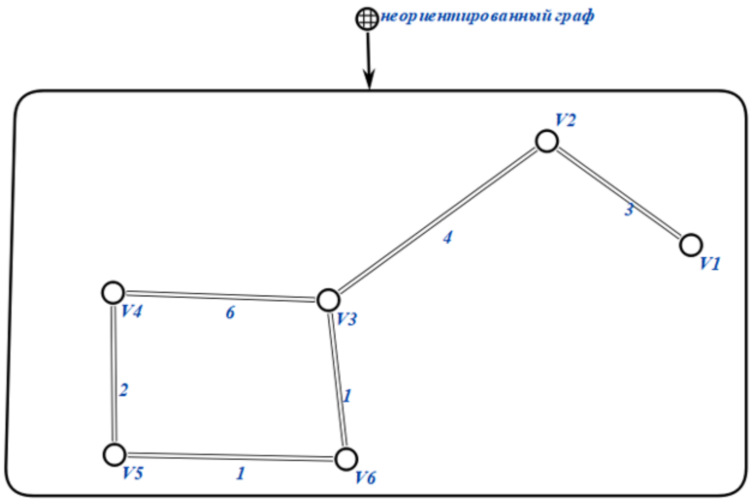

Рис. 13: Вход теста 4

Выход: Радиус графа равен 7.

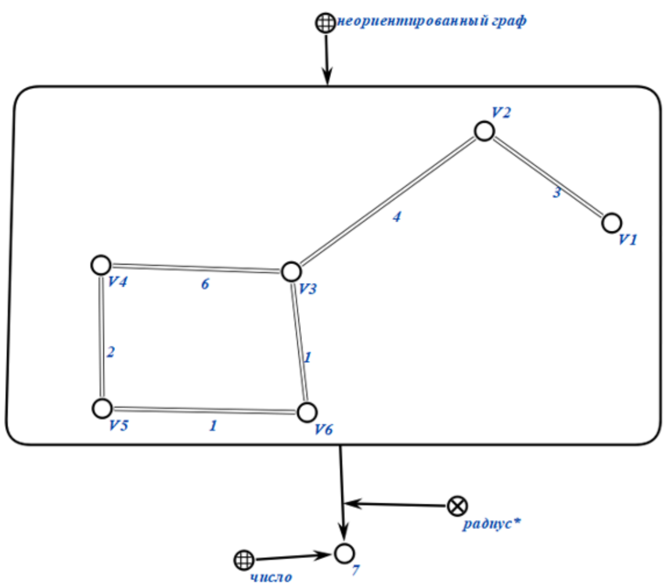

Рис. 14: Выход теста 4

3.5 Тест 5 (Рис.15, Рис.16)

Вход: Необходимо найти радиус взвешенного неориентированного графа.

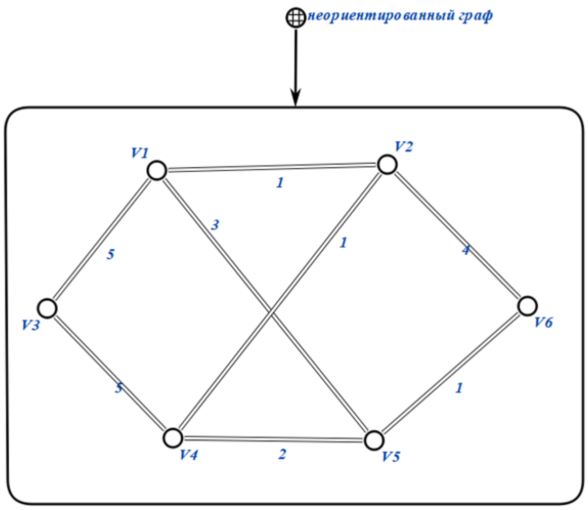

Рис. 15: Вход теста 5

Выход: Радиус графа равен 5.

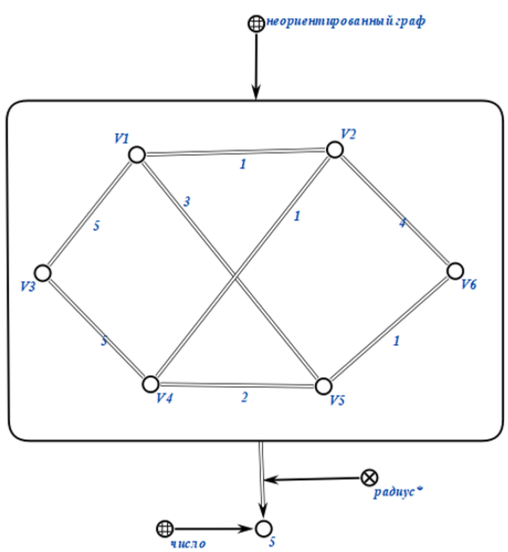

Рис. 16: Выход теста 5

4 Алгоритм

Для решения задачи необходимы следующие переменные:

1. Очередь *q* для хранения вершин, которые будут рассматриваться.
2. Начальная вершина *start*, для которой будем искать эксцентриситет.
3. Первая вершина *cur*, извлекаемая из очереди q.
4. *dist[V]* – самое короткое расстояние от данной вершины V графа до начальной вершины start.
5. *maxDist* – эксцентриситет вершины start.
6. *minEccentricity* – радиус графа.

Описание алгоритма:

1. Нахождение эксцентриситета для каждой вершины графа:

   1.1. Для каждой вершины Vi графа вычисляем эксцентриситет:

   1.1.1. В очередь q добавляется начальная вершина start

   1.1.2. Для каждой вершины графа dist[Vi] – это бесконечность, за исключением начальной вершины start, для которой dist[start] равно 0

   1.1.3. Обход вершин:

   1.1.3.1. Если очередь q не пустая, то извлекаем из очереди вершину cur

   1.1.3.2. Для каждой соседней для cur вершины выполняем:

   1.1.3.2.1. Если мы можем добраться до соседа более коротким путем, чем тот, который мы знали ранее, то есть dist[Vi]>(dist[cur]+вес ребра между вершинами cur и Vi)

   1.1.3.2.1.1. Обновляем расстояние, то есть dist[Vi]=(dist[cur]+вес ребра между вершинами cur и Vi)

   1.1.3.2.1.2. Вершина Vi , у которой обновилось значение, добавляется в очередь q

   1.1.4. Значение maxDist инициализируется 0

   1.1.5. Проходим по всем вершинам Vi графа и обновляем значение maxDist, если dist[Vi] больше текущего значения maxDist

2. Нахождение радиуса графа:

   2.1. Значение minEccentricity инициализируется infinity

   2.1.1. Проходим по всем вершинам Vi графа и обновляем значение minEccentricity, если эксцентриситет данной вершиы Vi меньше текущего значения minEccentricity

3. Алгоритм завершает работу.

5 Пример выполнения алгоритма в sc-памяти для графа из теста 3:

Для наглядности формализации опустим некоторые отношения (*).

1. Нахождение эксцентриситета для каждой вершины графа:
   1.1. Выбирается вершина V1 графа, которая будет начальной вершиной start, и добавляем ее в очередь q
   1.2. dist[V2] = dist[V3] = dist[V4] = infinity, dist[start] = 0 (Рис.17)

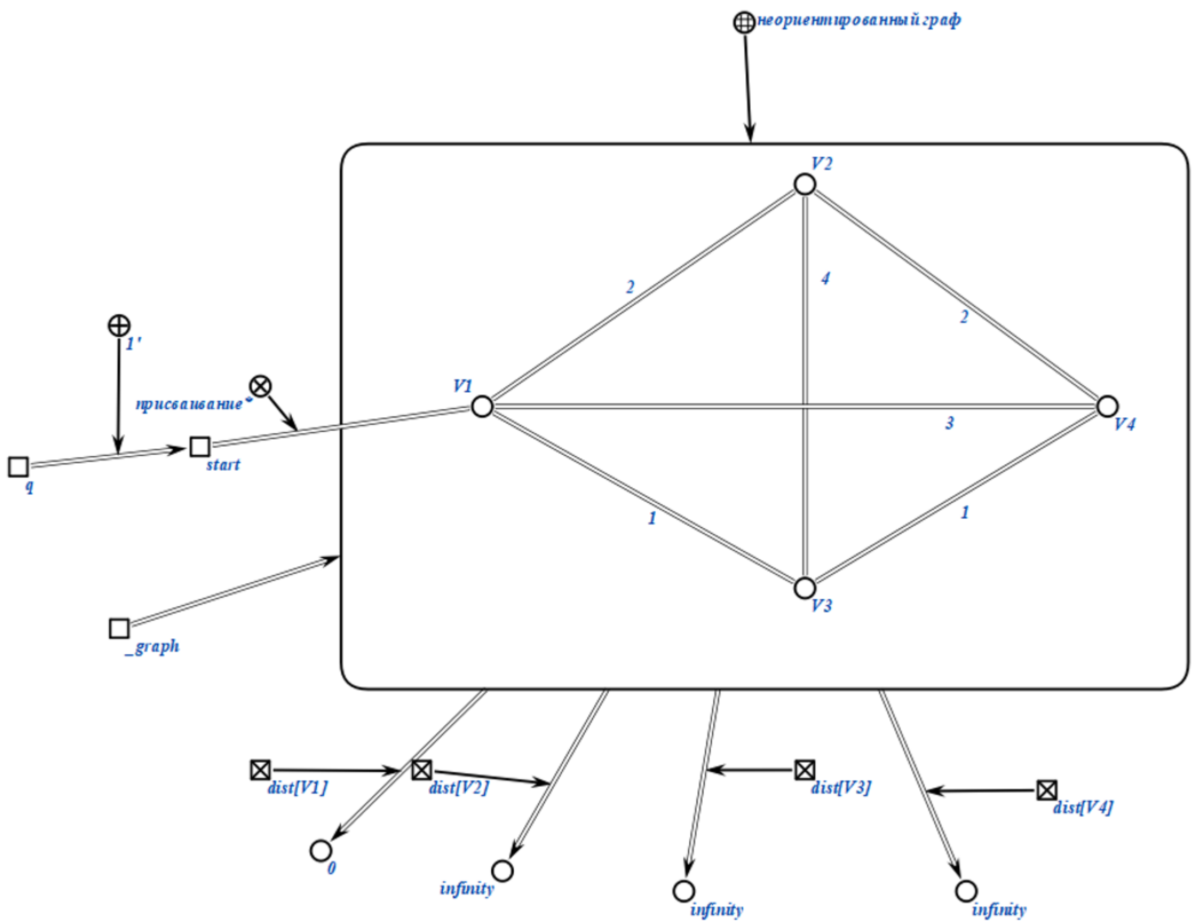

Рис. 17: Расстояние от V1 до V2, V3, V4

   1.3. Обход вершин:
      1.3.1. Извлекаем из очереди q вершину start=cur.
      1.3.2. Для каждой соседеней вершины V2, V3 и V4 выполняем:
         1.3.2.1. dist[V2]=infinity, (dist[cur]+вес ребра между вершинами cur и V2)=0+2=2, то есть infinity>2 (Рис.18)

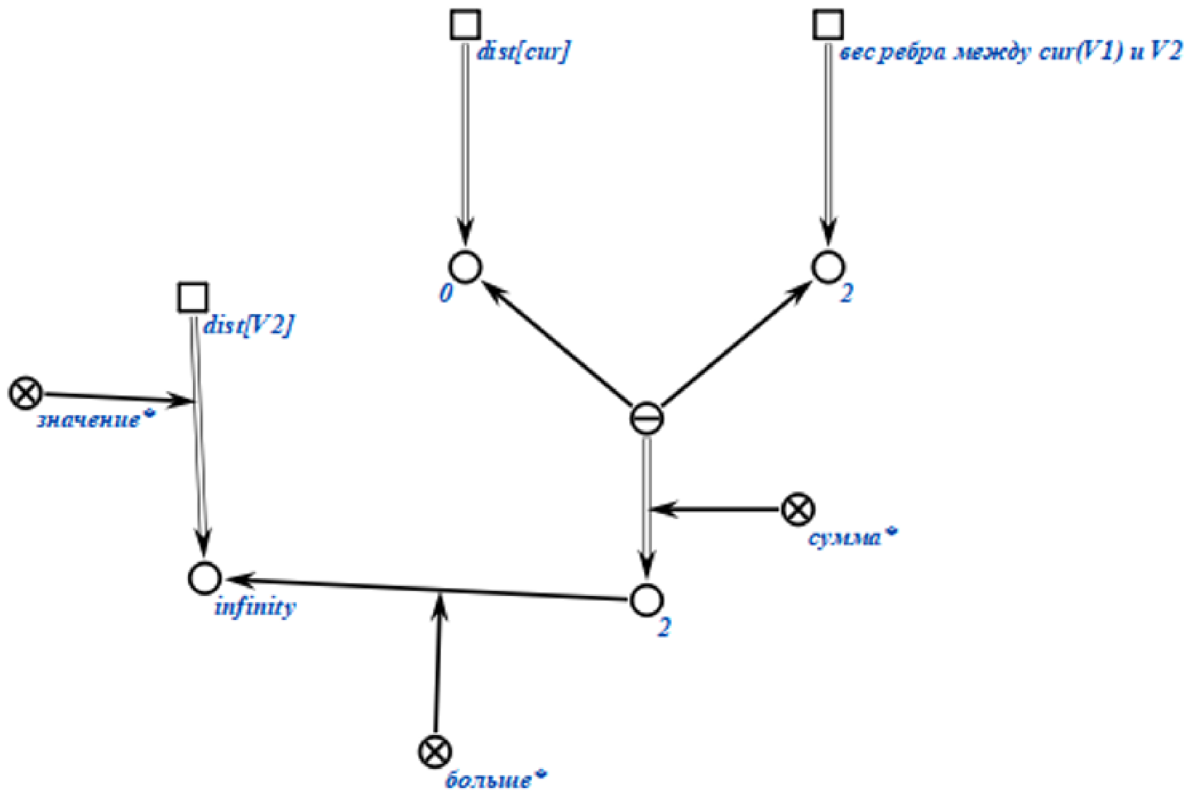

Рис. 18: Сравниваем dist[V2] и (dist[cur]+вес ребра между вершинами cur и V2)

1.3.2.2. Обновляем расстояние dist[V2]=2 (Рис.19)

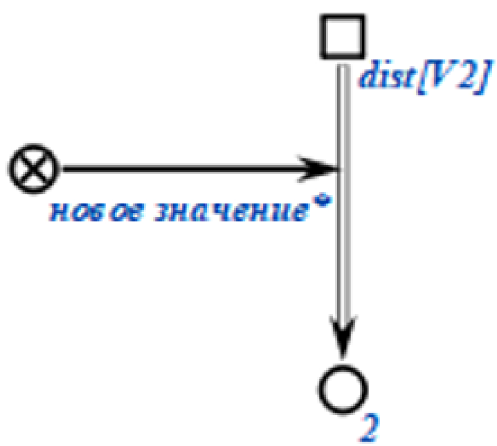

Рис. 19: Обнавление значения dist[V2]

1.3.2.3. Аналогично для V3 и V4 выпоним пункты 2.2.1 - 2.2.2 и получим dist[V3]=1, dist[V4]=3 (Рис.20)

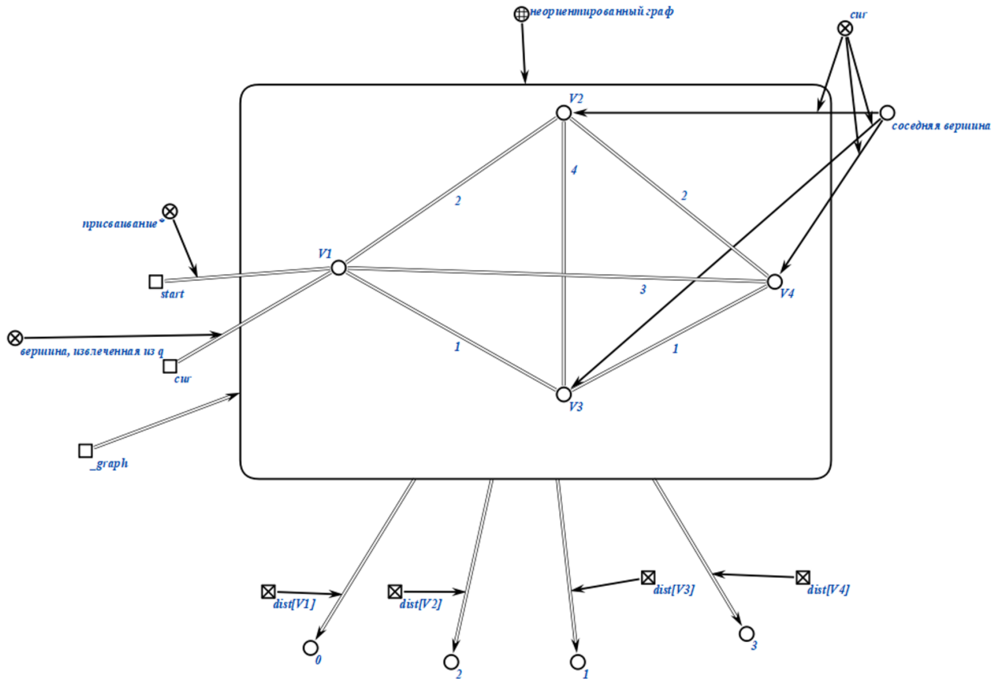

Рис. 20: Обнавление значений dist[V3], dist[V4]

1.3.2.4. Добавляем в очередь сначала V2, потом V3 и V4 (Рис.21)

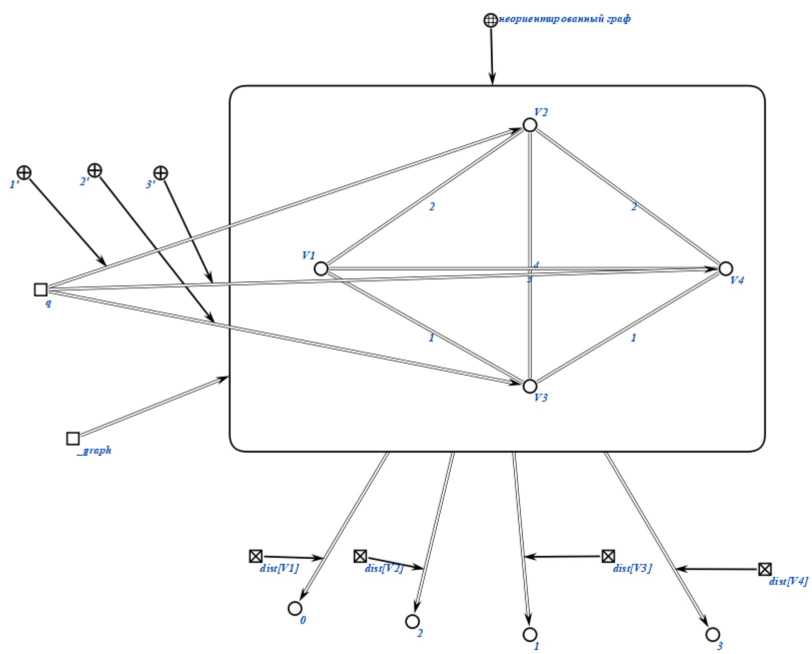

Рис. 21: Добавление в очередь V2, V3, V4

Выполняем пункт 1.3 пока очередь q не пуста. Результат показан на рисунке 22.

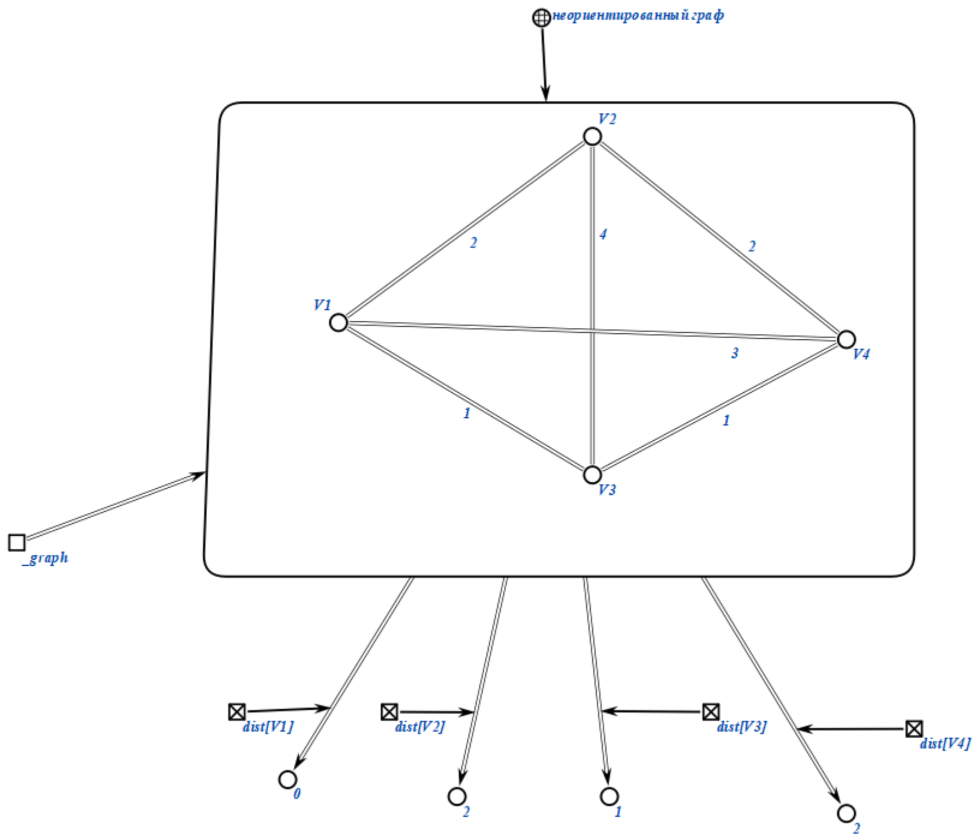

Рис. 22: Значения dist[V1], dist[V2], dist[V3], dist[V4] для V1

1.4. Значение maxDist инициализируется 0

1.5. Проходим по всем вершинам Vi графа и обновляем значение maxDist, если dist[Vi] больше текущего значения maxDist (Рис.23)

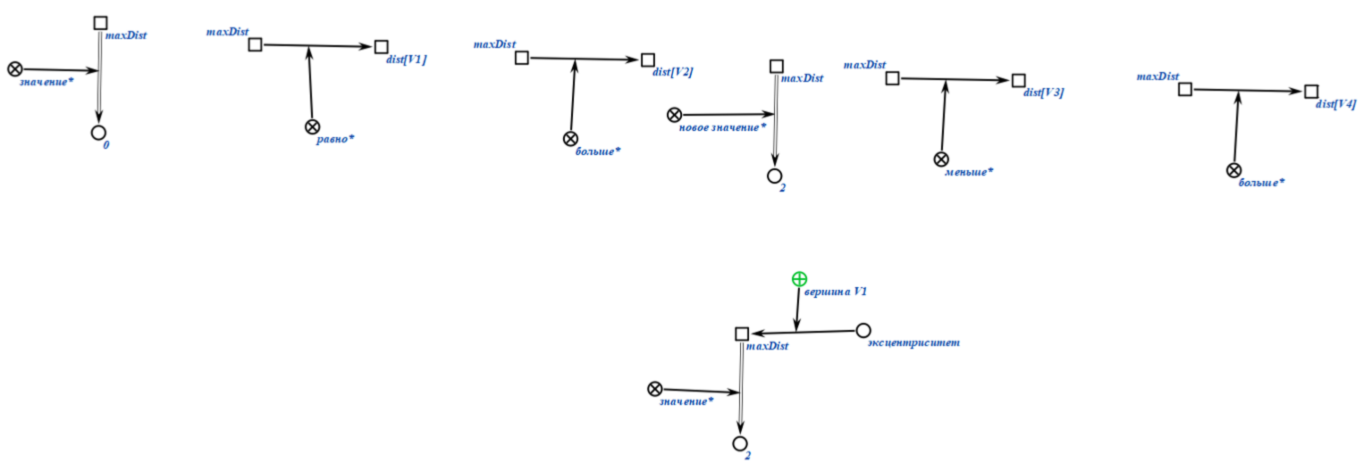

Рис. 23: Обновляем значение maxDist и находим эксцентриситет для V1

Пункт 1 повторим для оставшихся вершин V2, V3, V4. Результат показан на рисунке 24.

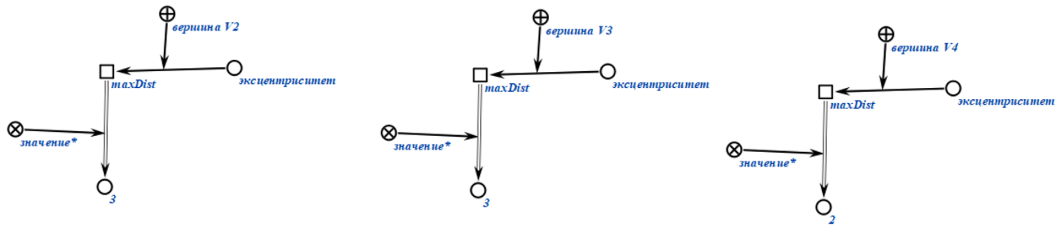

Рис. 24: Нахождение эксцентриситета для V2, V3, V4

2. Нахождение радиуса графа:
   2.1. Значение minEccentricity инициализируется infinity
   2.2. Проходим по всем вершинам Vi графа и обновляем значение minEccentricity, если эксцентриситет данной вершины Vi меньше текущего значения minEccentricity (Рис.25)

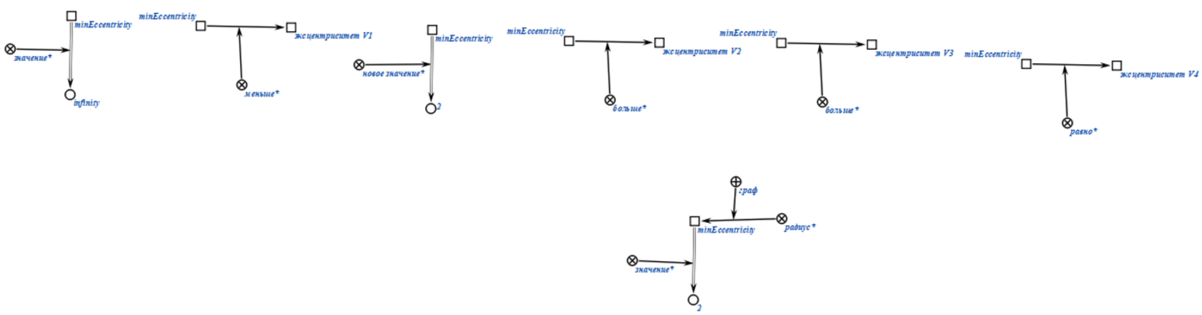

Рис. 25: Нахождение радиуса графа

3. Алгоритм завершает работу. На рисунке 26 показан результат выполения алгоритма для теста 3.

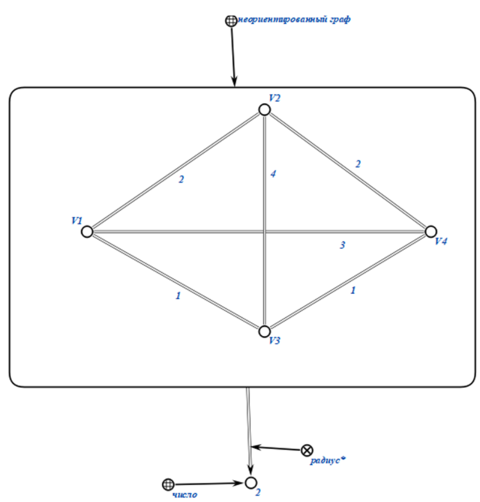

Рис. 26: Выход теста 3

Заключение

Продемонстрировала 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