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 «Построение фрагмента онтологии. Демонстрация работы программы решения теоретико-графовой задачи в семантической памяти»

Выполнила: Веркович Е.В.
студент гр. №221703

Проверил:
Загорский А.Г.

Минск 2023

Содержание

Введение

Цель: Получить навыки формализации и обработки информации с использованием семантических сетей. Разработать программу решения теоретико-графовой задачи на языке программирования.

Задача: Вариант 1.2: Определить вид графа: Ациклический граф (неориентированный граф, ориентированный граф)

Список используемых понятий

1. Граф (абсолютное понятие) - математическая абстракция реальной системы любой природы, объекты которой обладают парными связями. Граф как математический объект есть совокупность двух множеств — множества самих объектов, называемого множеством вершин, и множества их парных связей, называемого множеством рёбер. Элемент множества рёбер есть пара элементов множества вершин.

2. Простой граф (абсолютное понятие) - граф, в котором нет петель и кратных рёбер.

3. Ориентированный граф (абсолютное понятие) - в ориентированном графе ребра являются направленными, т.е. существует только одно доступное направление между двумя связными вершинами. Рёбра в орграфе также называют дугами.

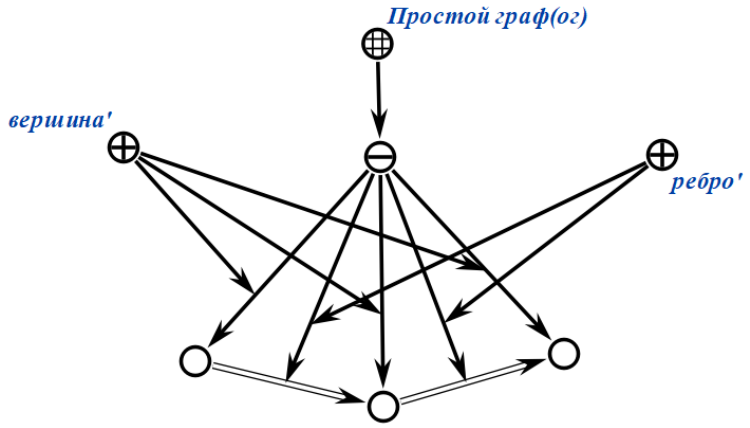


Рисунок 1: Формализация понятия "простой ориентированный граф"

4. Неориентированный граф (абсолютное понятие) - в неориентированном графе по каждому из ребер можно осуществлять переход в обоих направлениях.

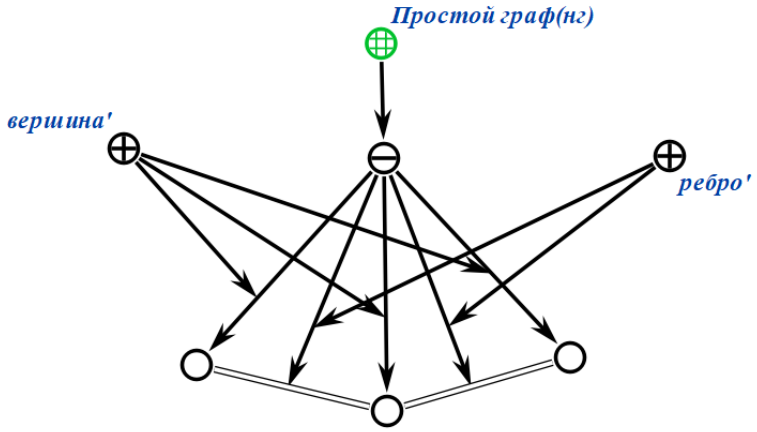


Рисунок 2: формализация понятия "простой неориентированный граф"

5. Вершина графа (относительное понятие) - точка в графе, отдельный объект, для топологической модели графа не имеет значения координата вершины, её расположение, цвет, вкус, размер; однако при решении некоторых задачах вершины могут раскрашиваться в разные цвета или сохранять числовые значения.

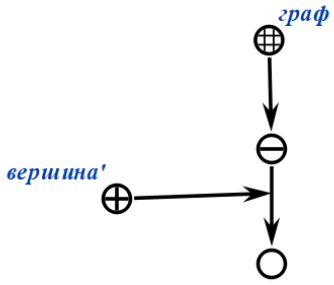


Рисунок 3: формализация понятия "вершина графа"

6. Ребро графа (относительное понятие) - неупорядоченная пара двух вершин, которые связаны друг с другом. Эти вершины называются концевыми точками или концами ребра. При этом важен сам факт наличия связи, каким именно образом осуществляется эта связь и по какой дороге - не имеет значения; однако рёбра может быть присвоен “вес”, что позволит говорить о “нагруженном графе” и решать задачи оптимизации.

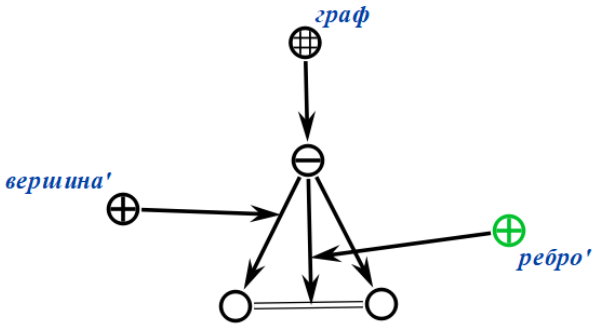


Рисунок 4: формализация понятия "ребро графа"

7. Смежность вершин (относительное понятие) - две вершины называются смежными, если они инцидентны одному ребру.

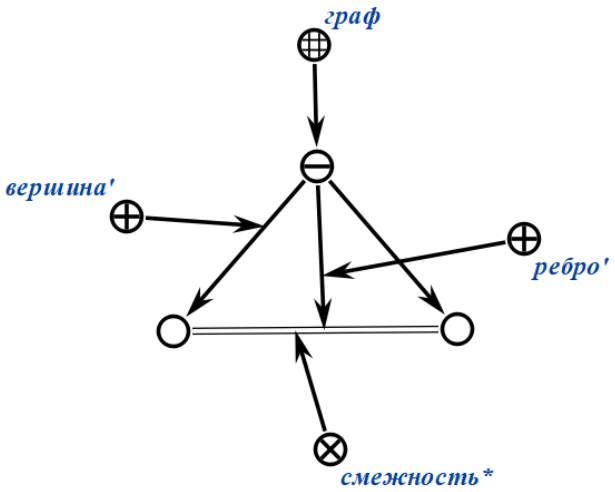


Рисунок 5: формализация понятия "смежность вершин"

8. Инцидентность (относительное понятие) – инцидентной называется вершина, являющаяся началом или концом ребра.

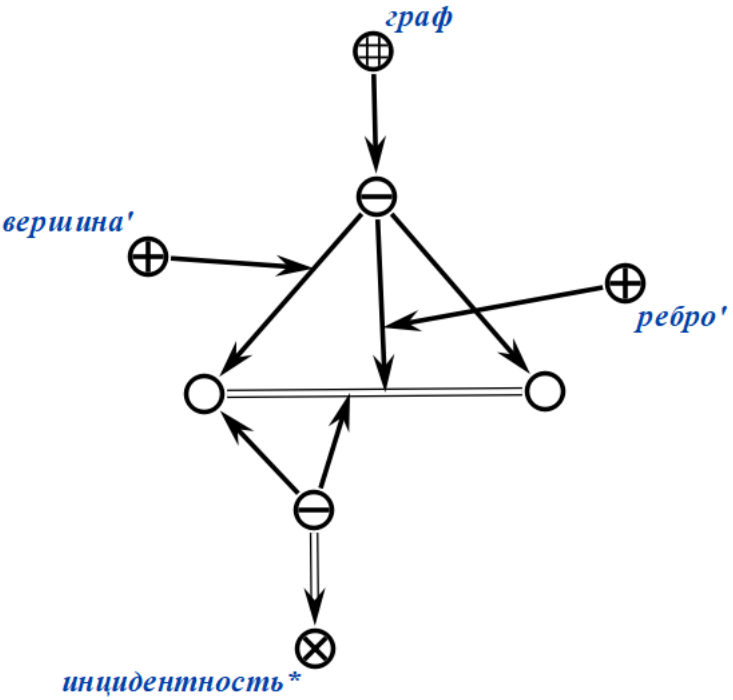


Рисунок 6: формализация понятия "инцидентность"

9. Путь (Цепь) - Путем или цепью в графе называют конечную последовательность вершин, в которой каждая вершина (кроме последней) соединена со следующей в последовательности вершин ребром

10. Цикл (относительное понятие) - цепь, в которой последняя вершина совпадает с первой.

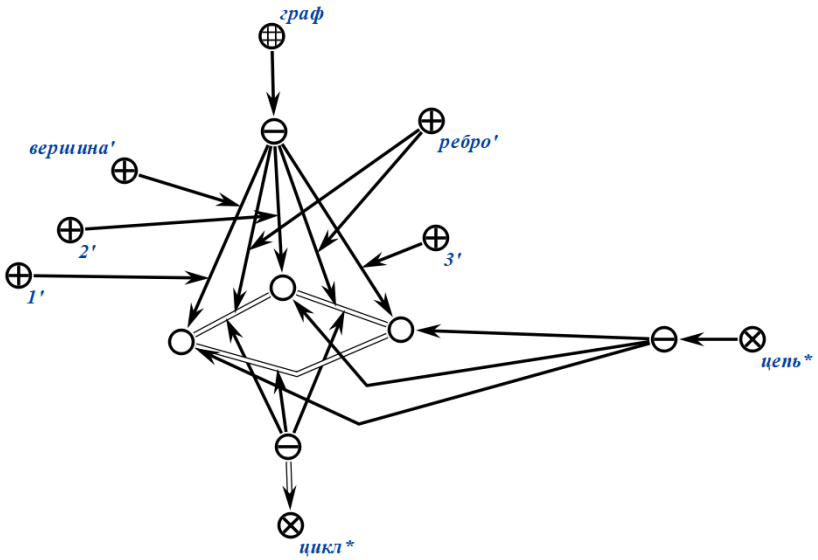


Рисунок 7: формализация понятий "путь"/"цепь" и "цикл"

11. Ациклический граф (абсолютное понятие) - граф не содержащий циклов.

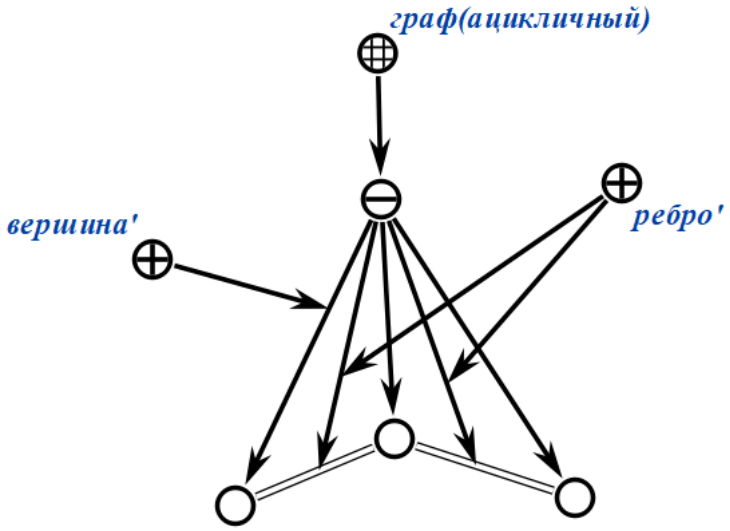


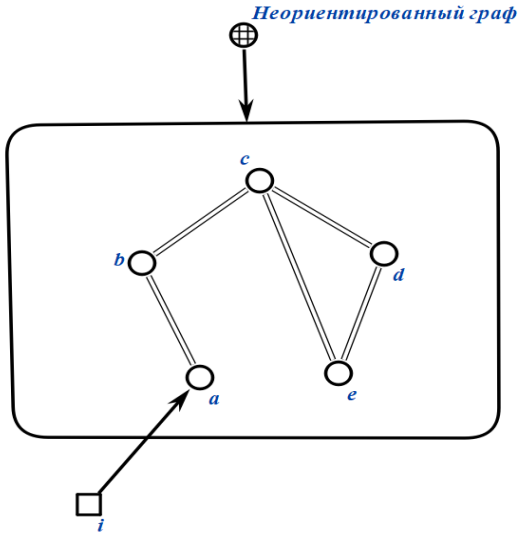
Рисунок 8: формализация понятия "ацикличный граф"

Описание алгоритма

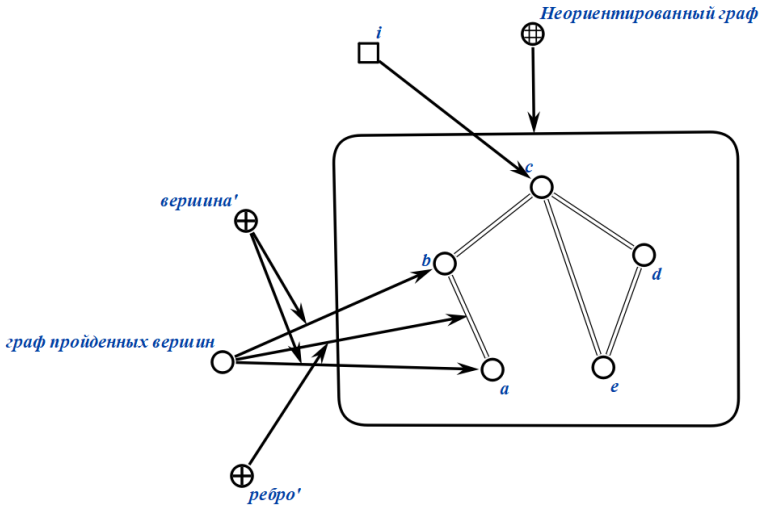
Чтобы проверить граф на ацикличность, т.е. на отсутствие циклов в нём, проведём серию поисков в глубину из каждой вершины, по которой ещё не прошёл алгоритм. При входе в вершину, она будет обозначаться, как пройденная, данную вершину поместим в граф посещённых вершин, таким образом запомним вхождение в неё. Если поиск будет пытаться пройти в пройденную вершину, значит цикл найден и, следовательно, граф цикличный, если такая ситуация не происходит – граф ацикличный.

Разберём алгоритм на примере простого неориентированного графа с циклом c-d-e-c.

1) Введём итератор «i», которым мы будем обозначать текущую вершину. Начнём процедуру с вершины «a».

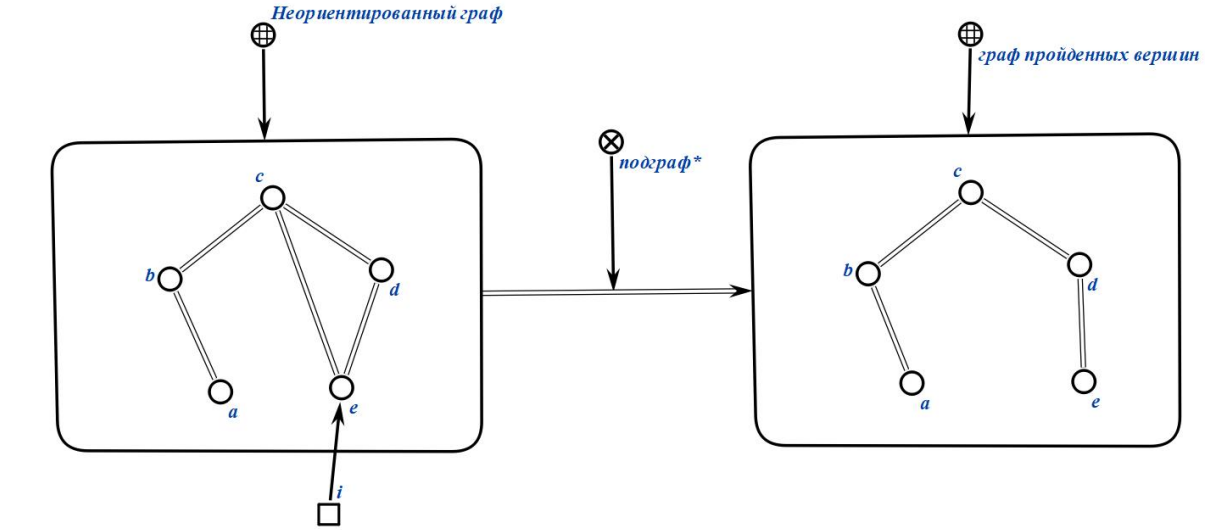


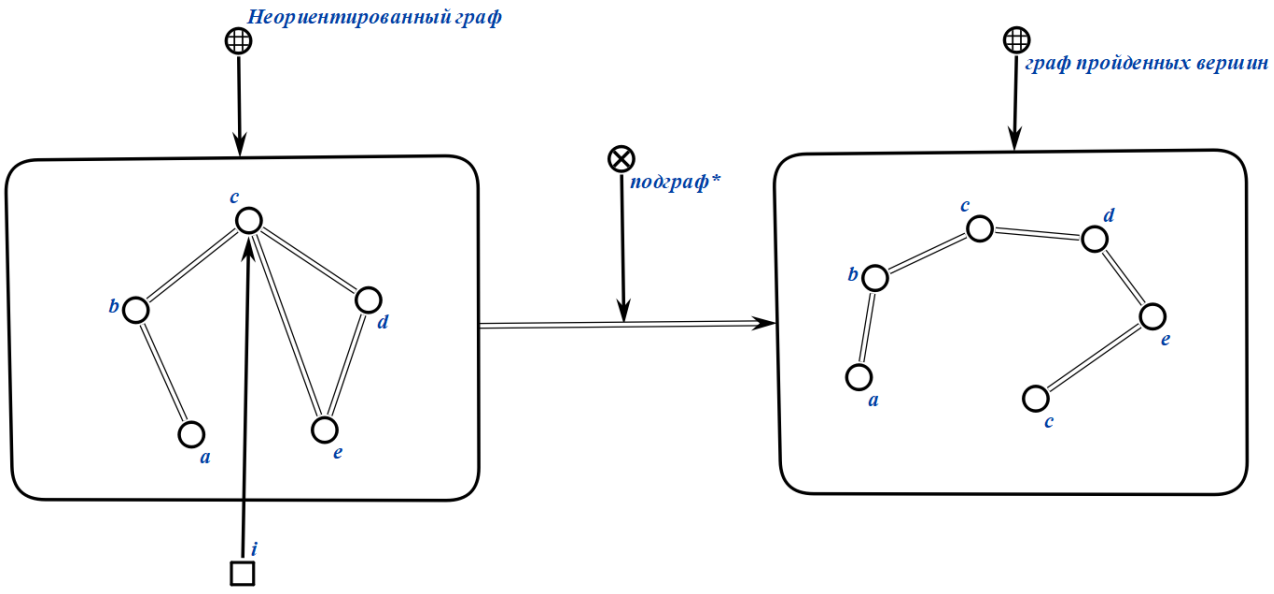
2) Далее заходим в следующую вершину «b», запомним «a» как пройденную и добавим в граф пройденных вершин.



Затем в вершину «c» и так далее.

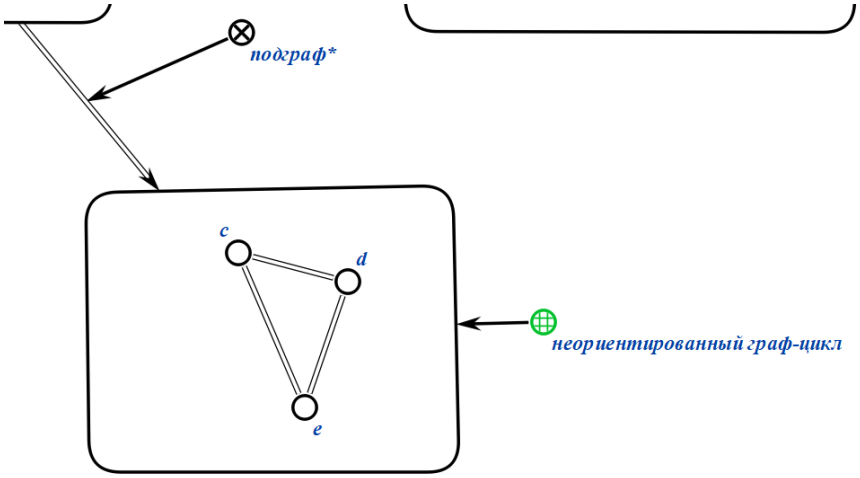
3) Теперь итератор стоит на вершине «e», от которой отходит ребро соединяющее её с вершиной «c». При следующем шаге итерации «i» будет находиться снова в вершине «c».





Таким образом в граф посещённых вершин добавится вершина «c». Видно, что в графе пройденных она уже присутствует, т.е. итератор дважды посетил одну вершину, значит в этом месте замкнулся цикл. А вид графа – циклический.

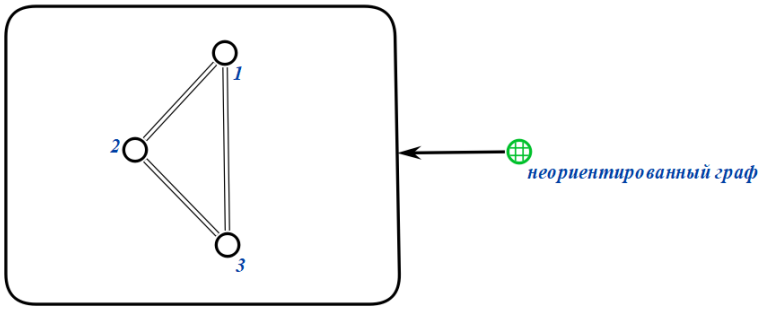
Подграф-цикл данного графа:



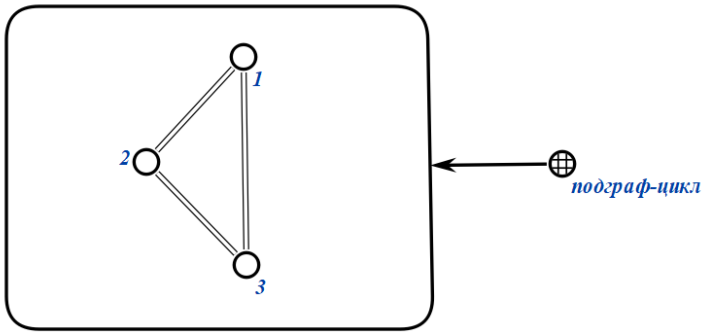
Тесты

Тест 1:

Input:

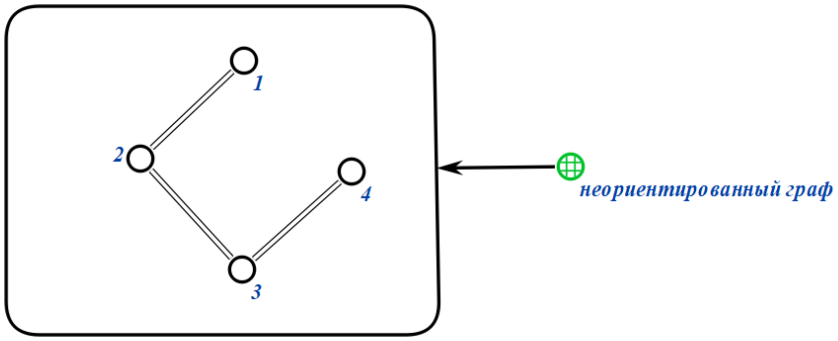


Output:

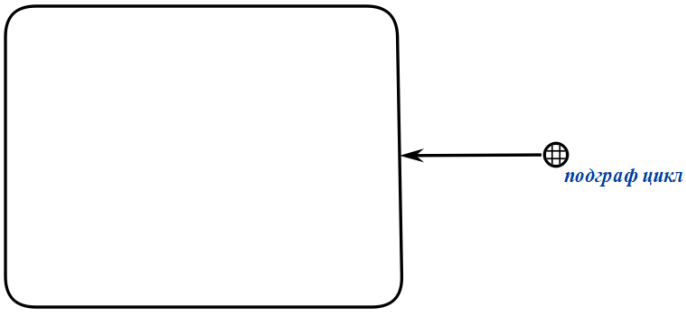


Тест 2:

Input:

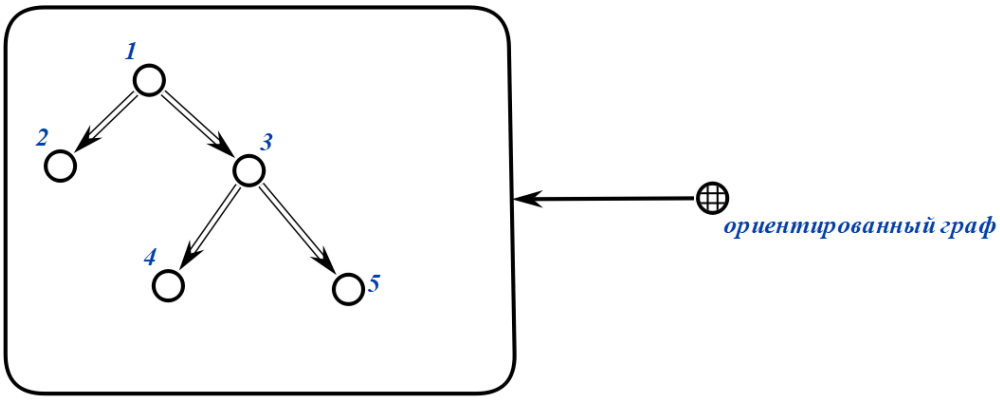


Output: граф ацикличный, значит и подграфов-циклов в нём нет.

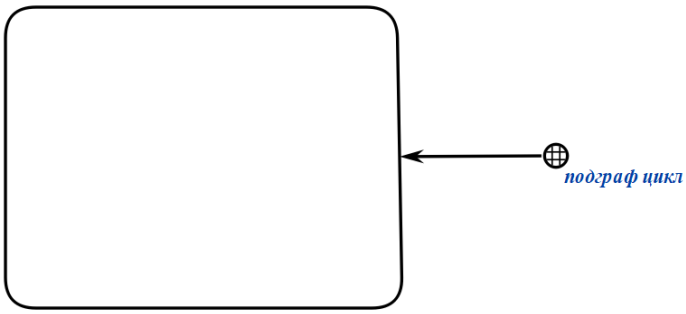


Тест 3:

Input:

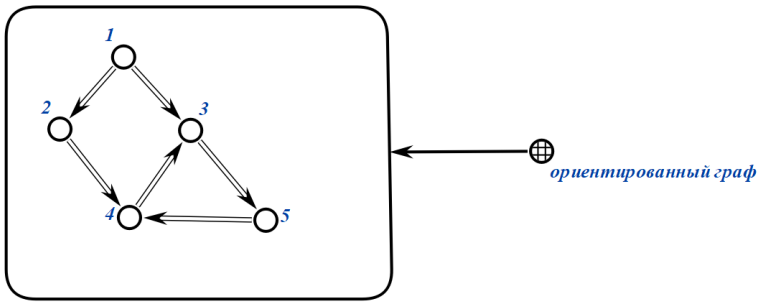


Output:

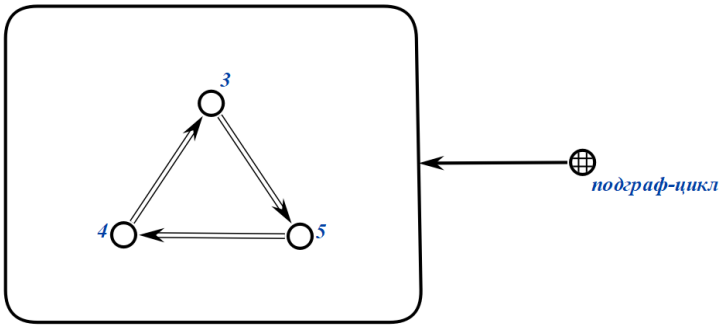


Тест 4:

Input:

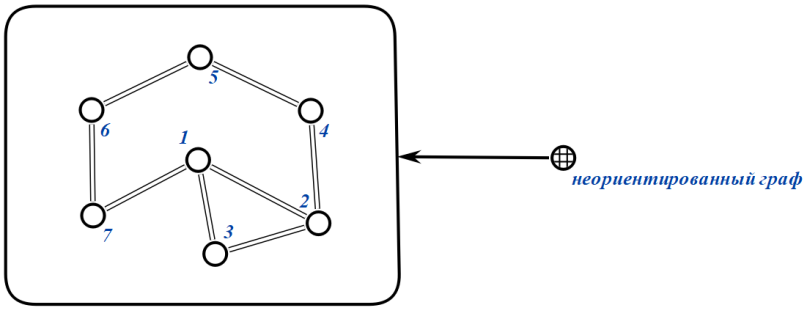


Output:

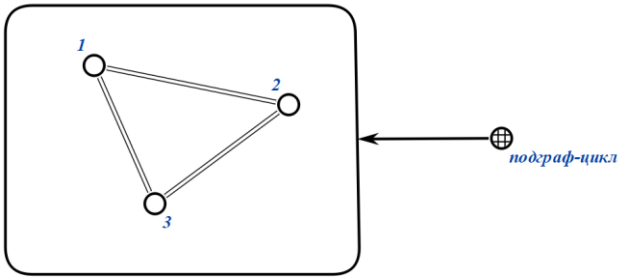


Тест 5:

Input:



Output:



Вывод

В результате данной расчётной работы я получила навыки формализации, закрепила теорию графов, реализовала программу определения вида графа.

Источники

- https://e-maxx.ru/index.php
- https://prog-cpp.ru/data-graph/#height
- https://drive.google.com/drive/folders/0B3Bkrx-aVpkSM3AtZ0Zub3JYUG8?resourcekey=0-2ScXl9Cg3IQQmH0oU483qA
- https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)#%D0%9F%D1%80%D0%BE%D1%81%D1%82%D0%BE%D0%B9_%D0%B3%D1%80%D0%B0%D1%84
- https://habr.com/ru/company/otus/blog/568026/
- http://shujkova.ru/sites/default/files/algorithm2.pdf
- https://metanit.com/