

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
Задача поиска реберного графа для неориентированного графа

Выполнил
студент группы
021704

Солдатенков Е.Д.

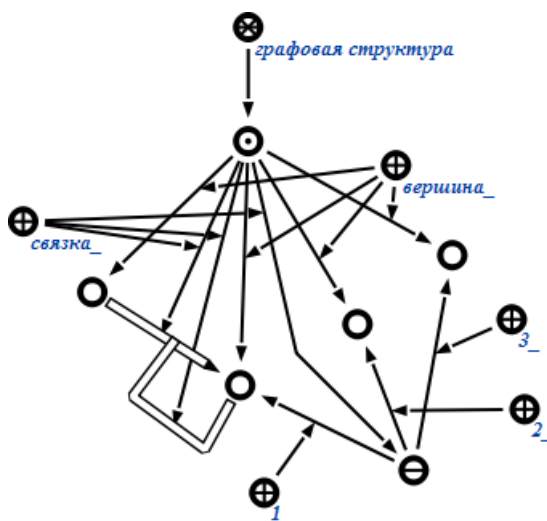
Проверил

Витязь В.С

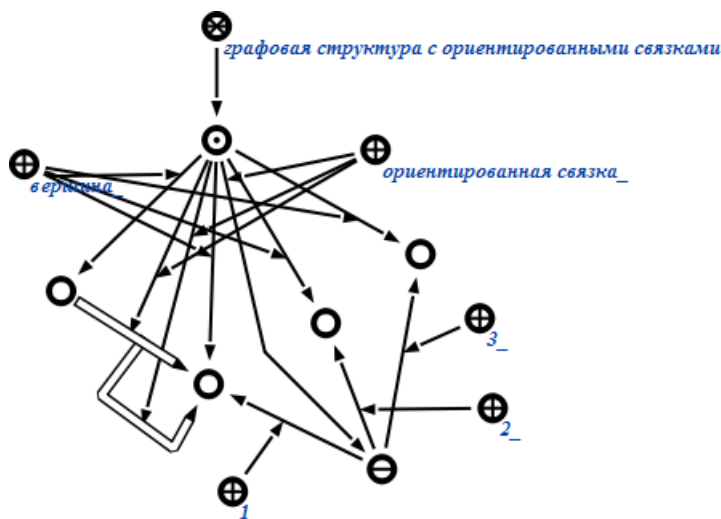
Цель: Получить навыки формализации и обработки информации с использованием семантических сетей

Задача: поиск реберного графа для неориентированного графа Список понятий

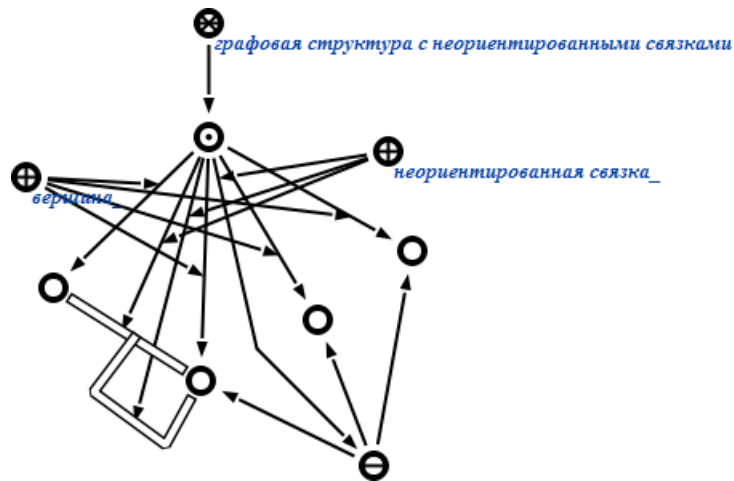
1. Графовая структура (абсолютное понятие) - это такая одноуровневая реляционная структура, объекты которой могут играть роль либо вершины, либо связки:
 - a. Вершина (относительное понятие, ролевое отношение);
 - b. Связка (относительное понятие, ролевое отношение).



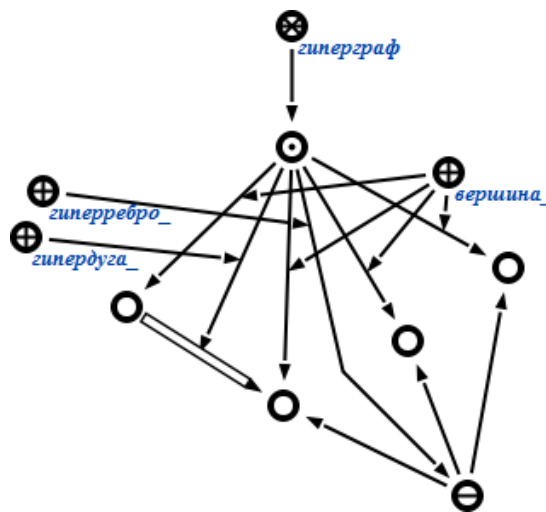
2. Графовая структура с ориентированными связками (абсолютное понятие)
 - a. Ориентированная связка (относительное понятие, ролевое отношение) – связка, которая задается ориентированным множеством.



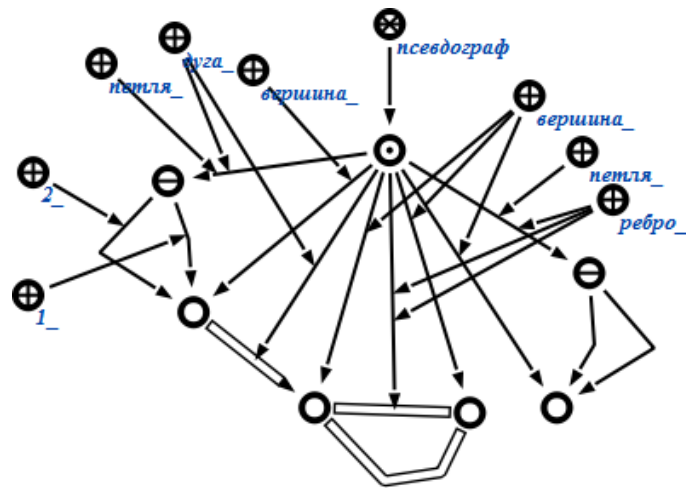
3. Графовая структура с неориентированными связками (абсолютное понятие)
 - a. Неориентированная связка (относительное понятие, ролевое отношение) – связка, которая задается неориентированным множеством.



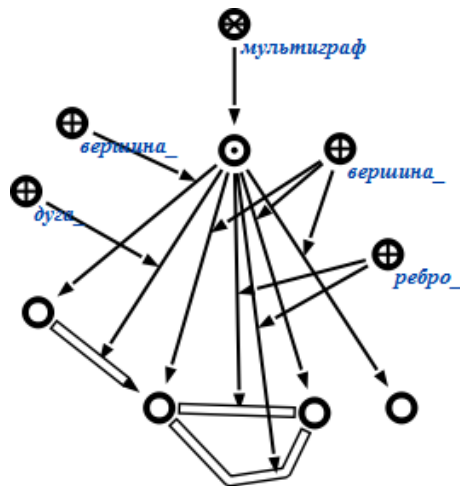
4. Гиперграф (абсолютное понятие) – это такая графовая структура, в которой связки могут связывать только вершины:
- Гиперсвязка (относительное понятие, ролевое отношение);
 - Гипердуга (относительное понятие, ролевое отношение) – ориентированная гиперсвязка;
 - Гиперребро (относительное понятие, ролевое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



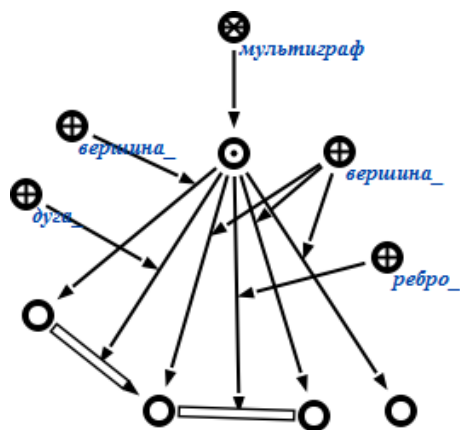
5. Псевдограф (абсолютное понятие) – это такой гиперграф, в котором все связки должны быть бинарными:
- Бинарная связка (относительное понятие, ролевое отношение) – гиперсвязка арности 2;
 - Ребро (относительное понятие, ролевое отношение) – неориентированная гиперсвязка;
 - Дуга (относительное понятие, ролевое отношение) – ориентированная гиперсвязка;
 - Петля (относительное понятие, ролевое отношение) – бинарная связка, у которой первый и второй компоненты совпадают.



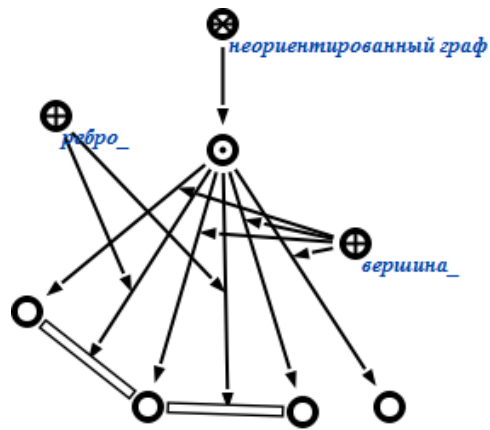
6. Мультиграф (абсолютное понятие) – это такой псевдограф, в котором не может быть петель:



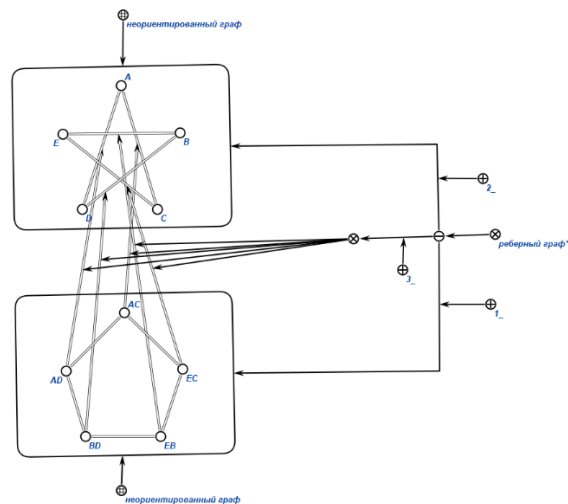
7. Граф (абсолютное понятие) – это такой мультиграф, в котором не может быть кратных связей, т.е. связей у которых первый и второй компоненты совпадают:



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



9. Реберный граф (относительное понятие) – это граф для неориентированного графа, представляющий собой соседство ребер основного графа. Для примера ниже приведен реберный граф.



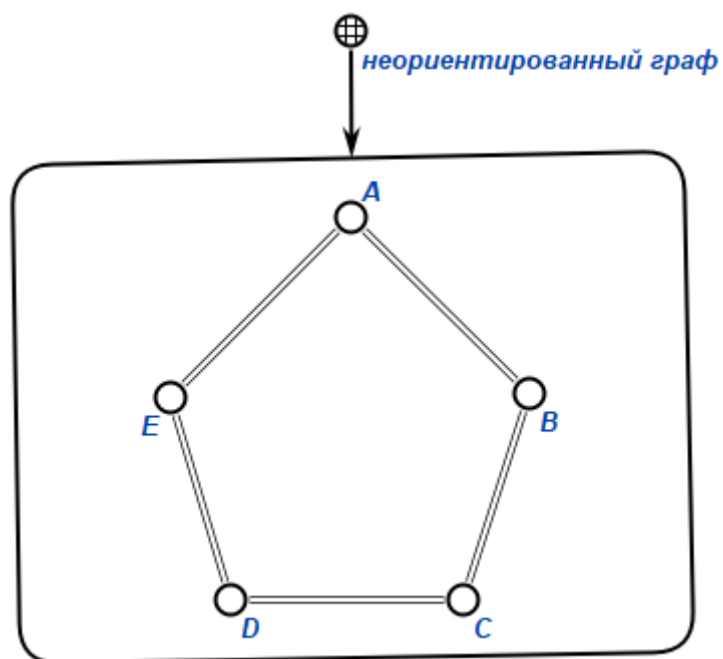
1 Тестовые примеры

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

1.1 Тест 1

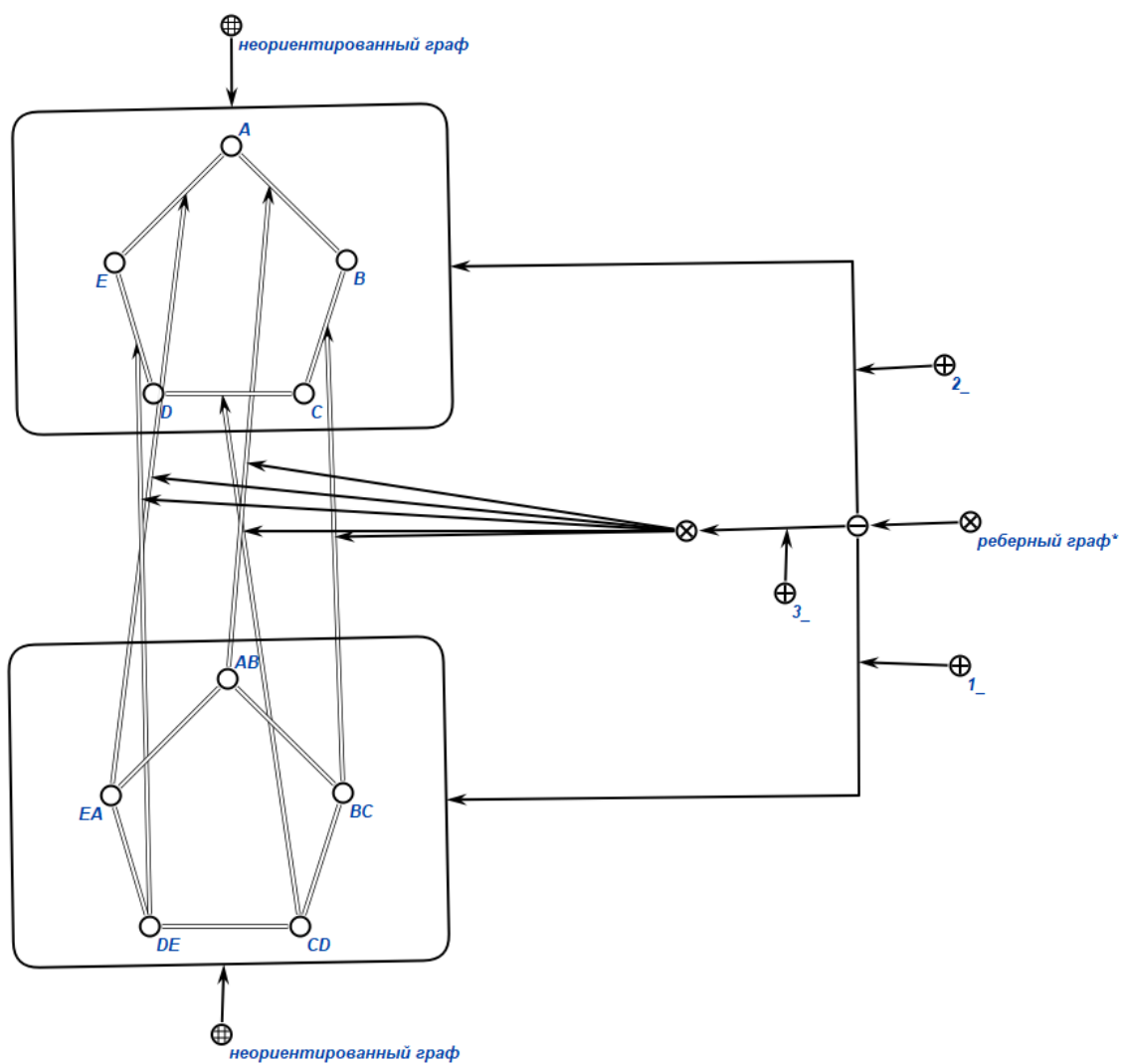
Вход:

Необходимо найти реберный граф для данного неориентированного графа.



Выход:

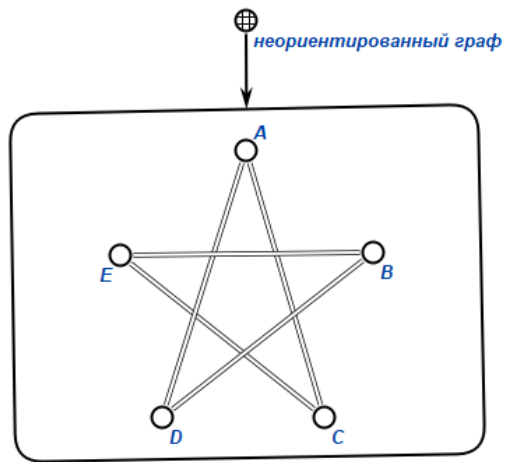
Будет найден граф $G = \langle V, E \rangle$, где $V = \{AB, BC, CD, DE, AE\}$.



1.2 Тест 2

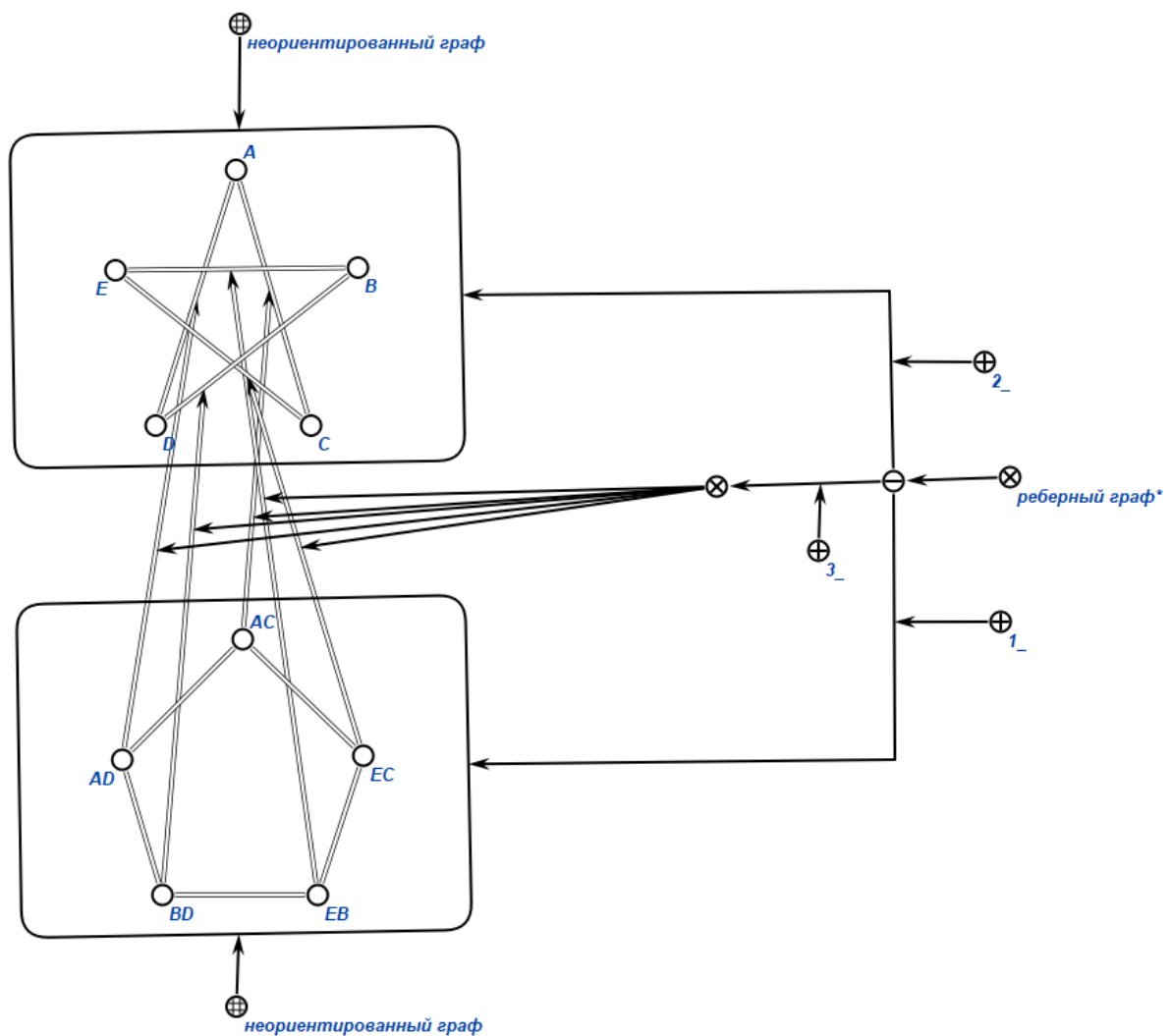
Вход:

Необходимо найти реберный граф для данного неориентированного графа.



Выход:

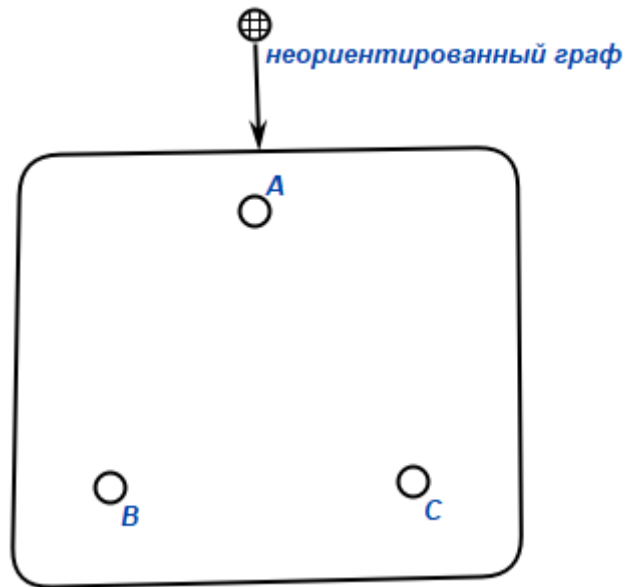
Будет найден граф $G = \langle V, E \rangle$, где $V = \{AC, EC, EB, DB, AD\}$.



1.3 Тест 3

Вход:

Необходимо найти реберный граф для данного неориентированного графа.



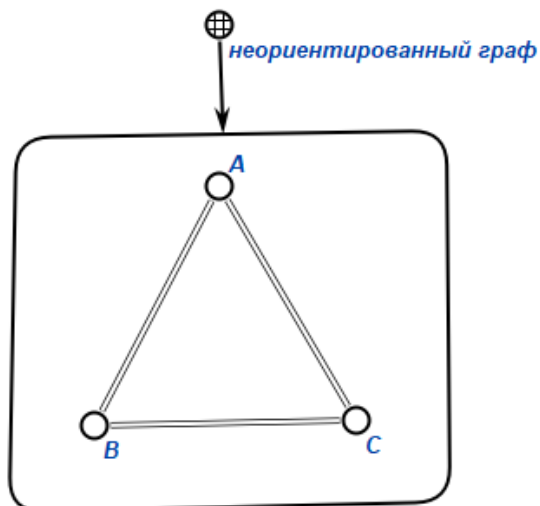
Выход:

Реберный граф не существует. Программа должна вернуть ошибку вызывающему контексту.

1.4 Тест 4

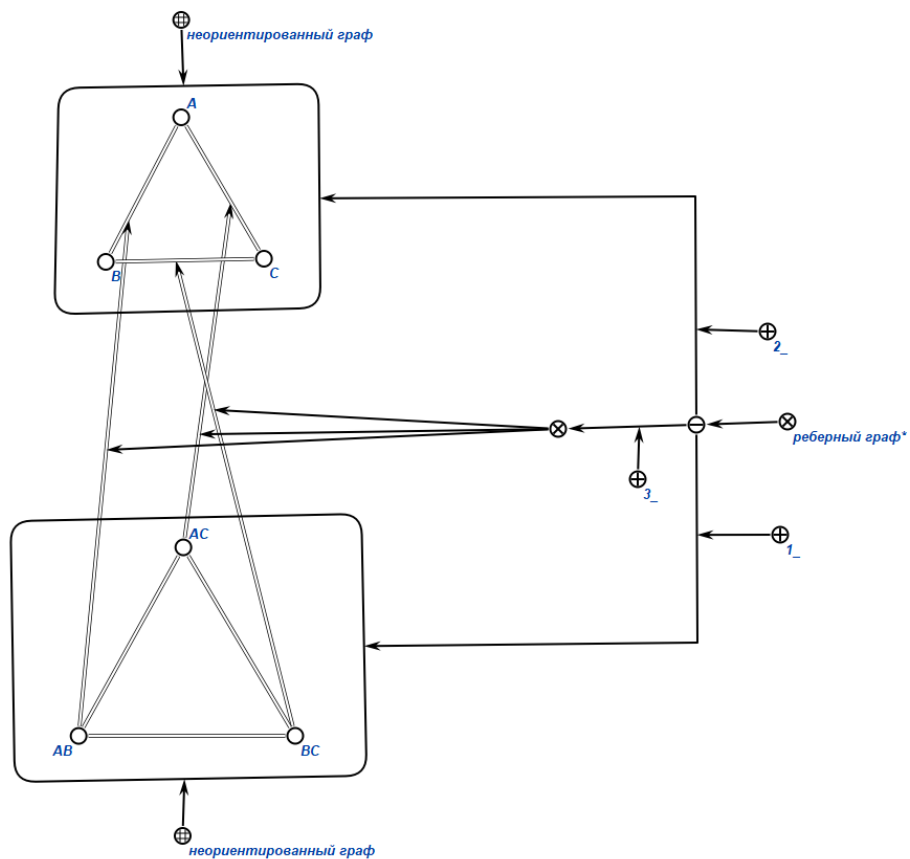
Вход:

Необходимо найти реберный граф для данного неориентированного графа.



Выход:

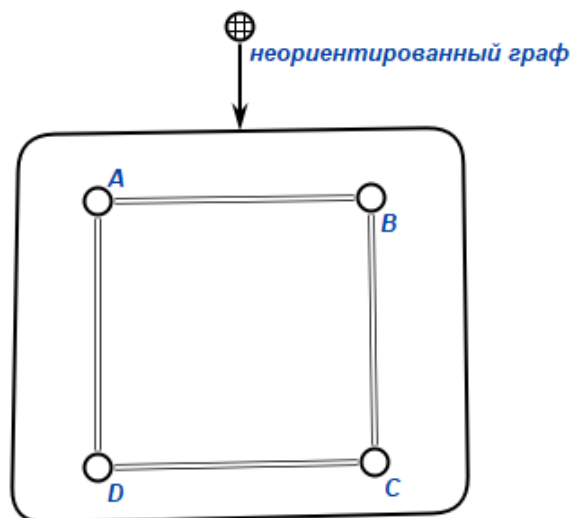
Будет найден граф $G = \langle V, E \rangle$, где $V = \{AC, AB, BC\}$.



1.5 Тест 5

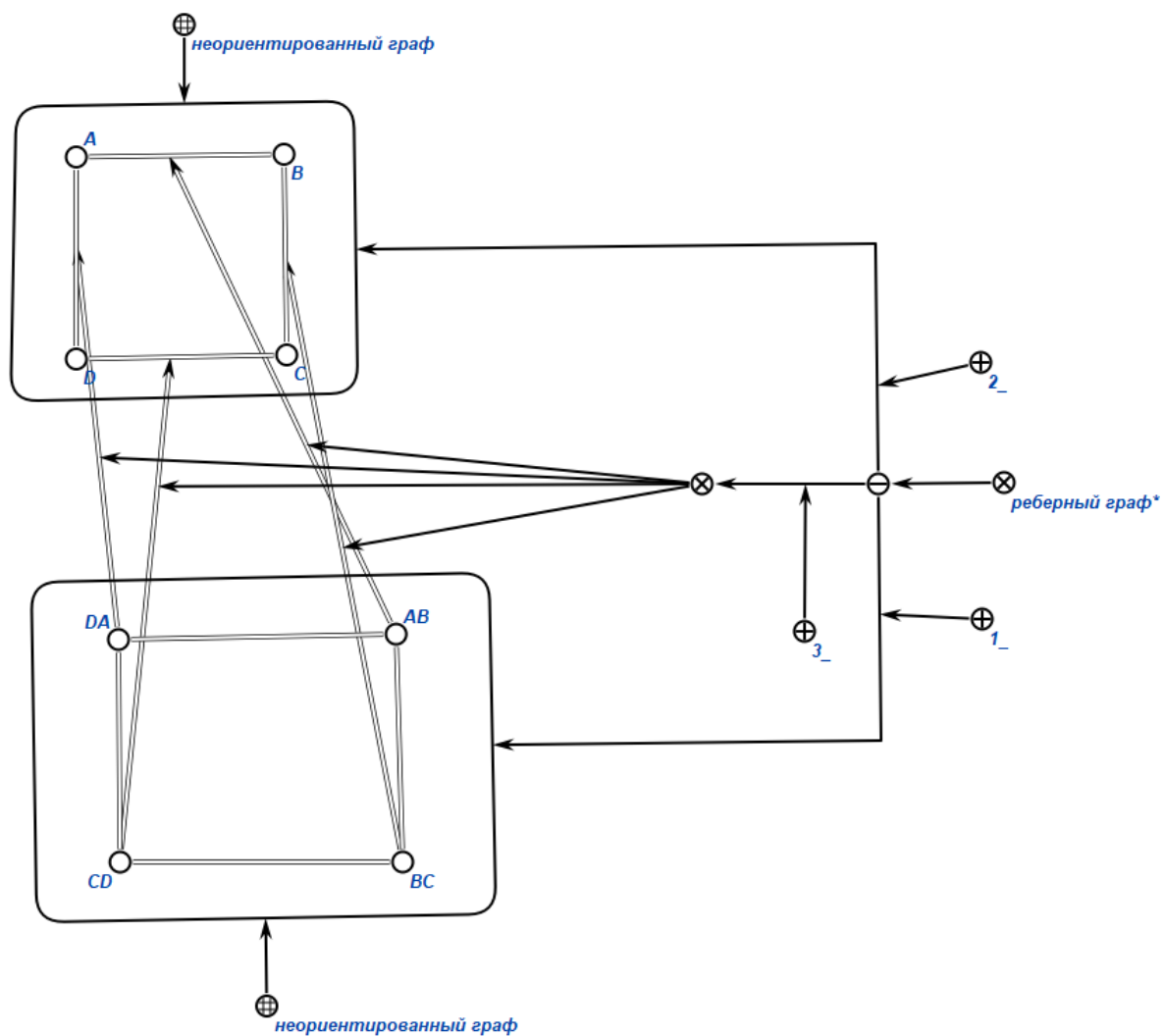
Вход:

Необходимо найти найти реберный граф для данного неориентированного графа



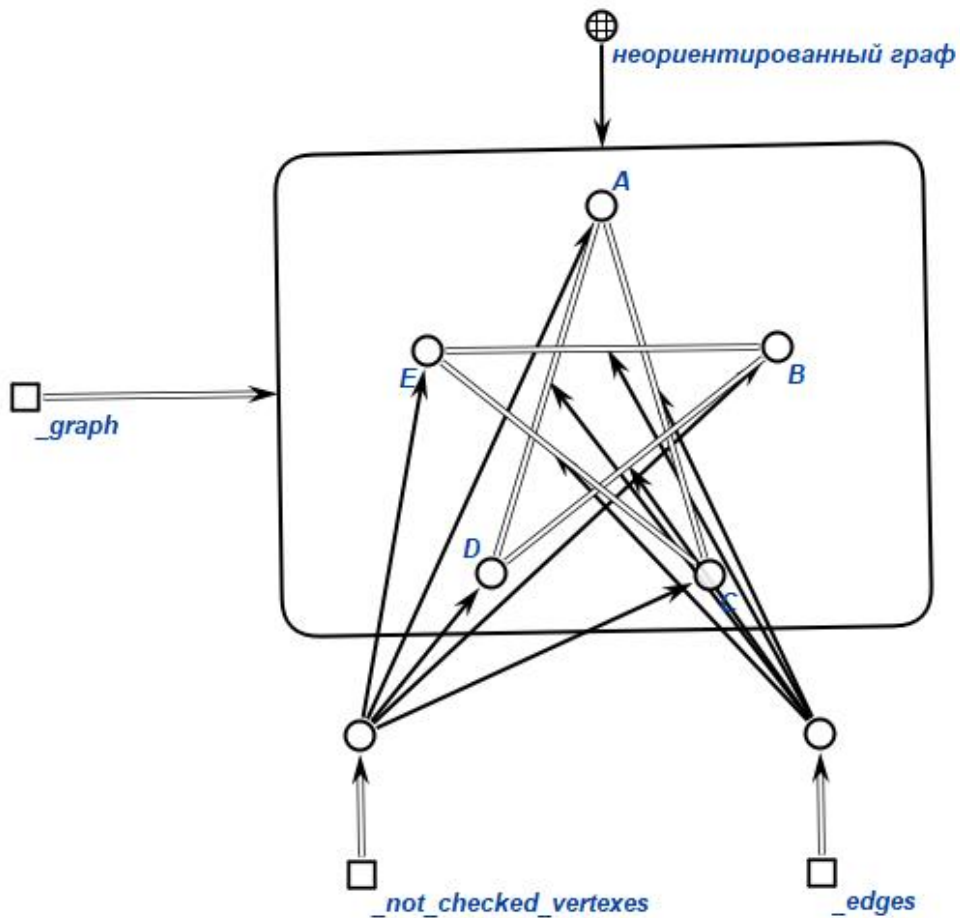
Выход:

Будет найден граф $G = \langle V, E \rangle$, где $V = \{DA, AB, BC, CD\}$.



2 Описание алгоритма

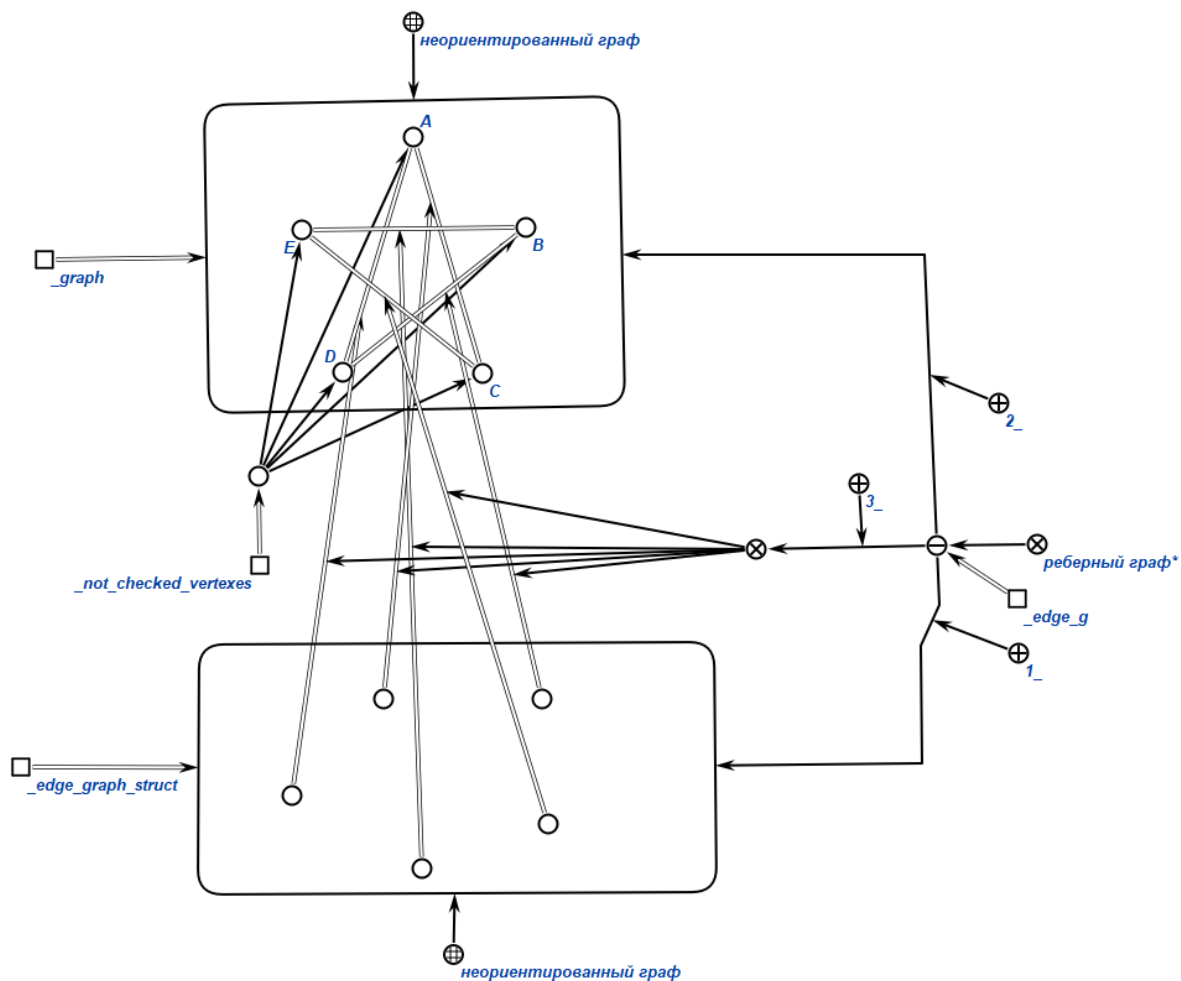
1. Задание входного графа, неориентированных множеств неиспользованных вершин ребер



Переменные изменятся следующим образом:

- `_graph` получит в качестве значения sc-узел неориентированного графа;
- `_not_checked_vertexes` получит в качестве значения множество непроверенных вершин обрабатываемого графа. Каждый раз нам следует проверять, остались ли еще неиспользованные вершины, является ли следующая в рассмотрении вершина неиспользованной.
- `_edges` получит в качестве значения множество ребер обрабатываемого графа.

2. Создание связки отношения реберного графа*



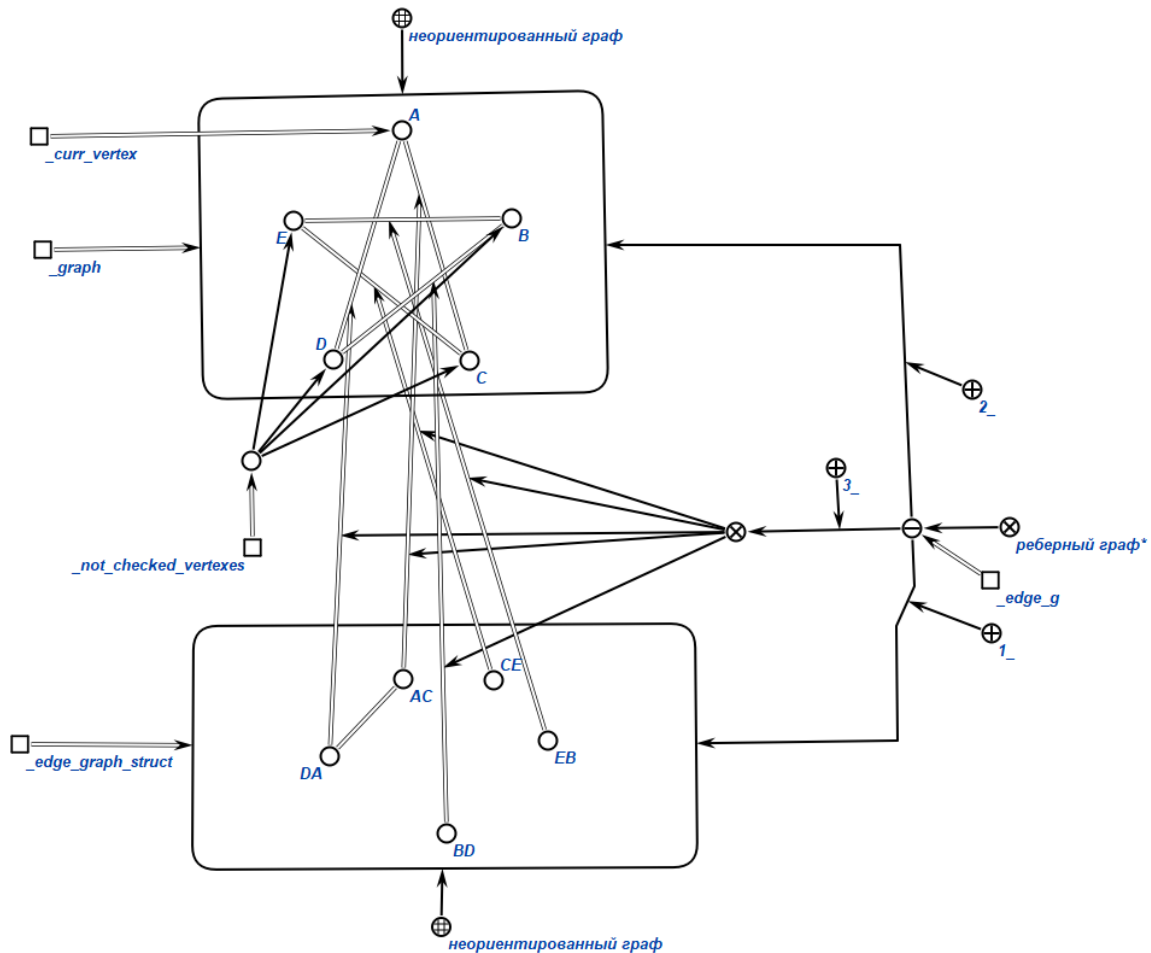
Начинаем генерацию реберного графа.

Создадим связку отношения реберного графа* и установим ее в качестве значения переменной `_edge_g`.

Переменная `_edge_graph_struct` получит в качестве значения неориентированный граф.

В созданном неориентированном графе создаем вершины, соответствующие каждому ребру основного графа (каждой вершине из множества `_edges`) и удаляем множество `_edges`. Указываем соответствия между ребрами основного и вершинами искомого графа.

3. Проверка ребер вершины A и создание ребра в реберном графе



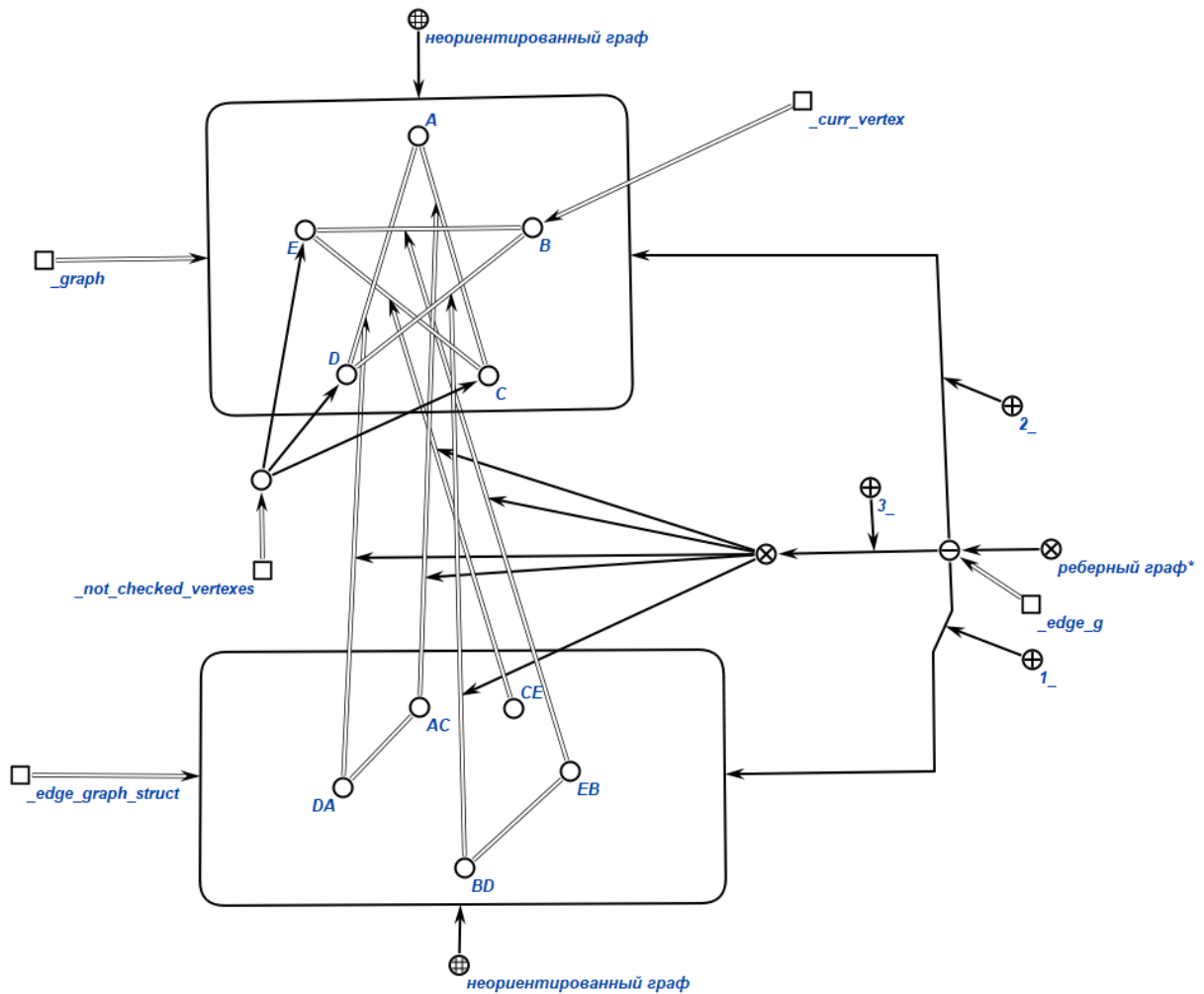
Переменная `_curr_vertex` получит в качестве значения первую вершину из множества неиспользованных вершин – A.

Начинаем исследование ребер вершины A.

Из вершины выходит два ребра: AC и DA. В реберном графе создаем ребро между вершинами DA и AC (для удобства назвали вершины в реберном графе названиями, соответствующими ребрам, связывающим вершины в основном графе). В дальнейшем, если вершина имеет только одно ребро, для этого ребра не будет создано новых связей в реберном графе.

Удаляем вершину A из множества неиспользованных вершин `_not_checked_vertices`.

4. Проверка ребер вершины В и создание ребра в реберном графе



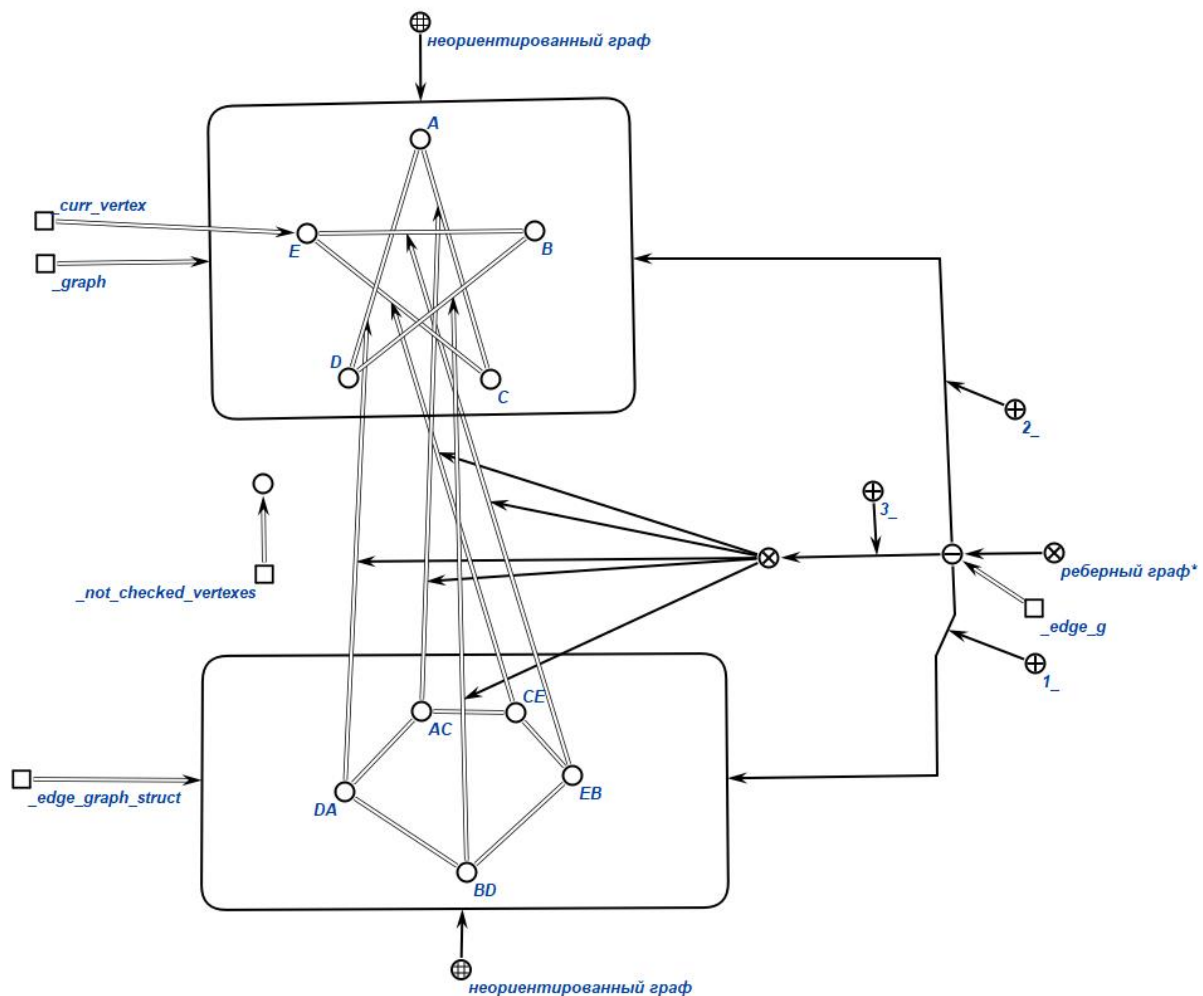
По аналогии с шагом 3 действуем с вершиной В.

Из вершины выходит два ребра: BD и BE. В реберном графе создаем ребро между вершинами DB и BE.

Удаляем вершину В из множества неиспользованных вершин.

По аналогии с шагом 3 действуем со всеми вершинами, пока множество непроверенных вершин не пусто.

5. Проверка ребер всех вершин из множества непроверенных вершин и создание ребер в реберном графе



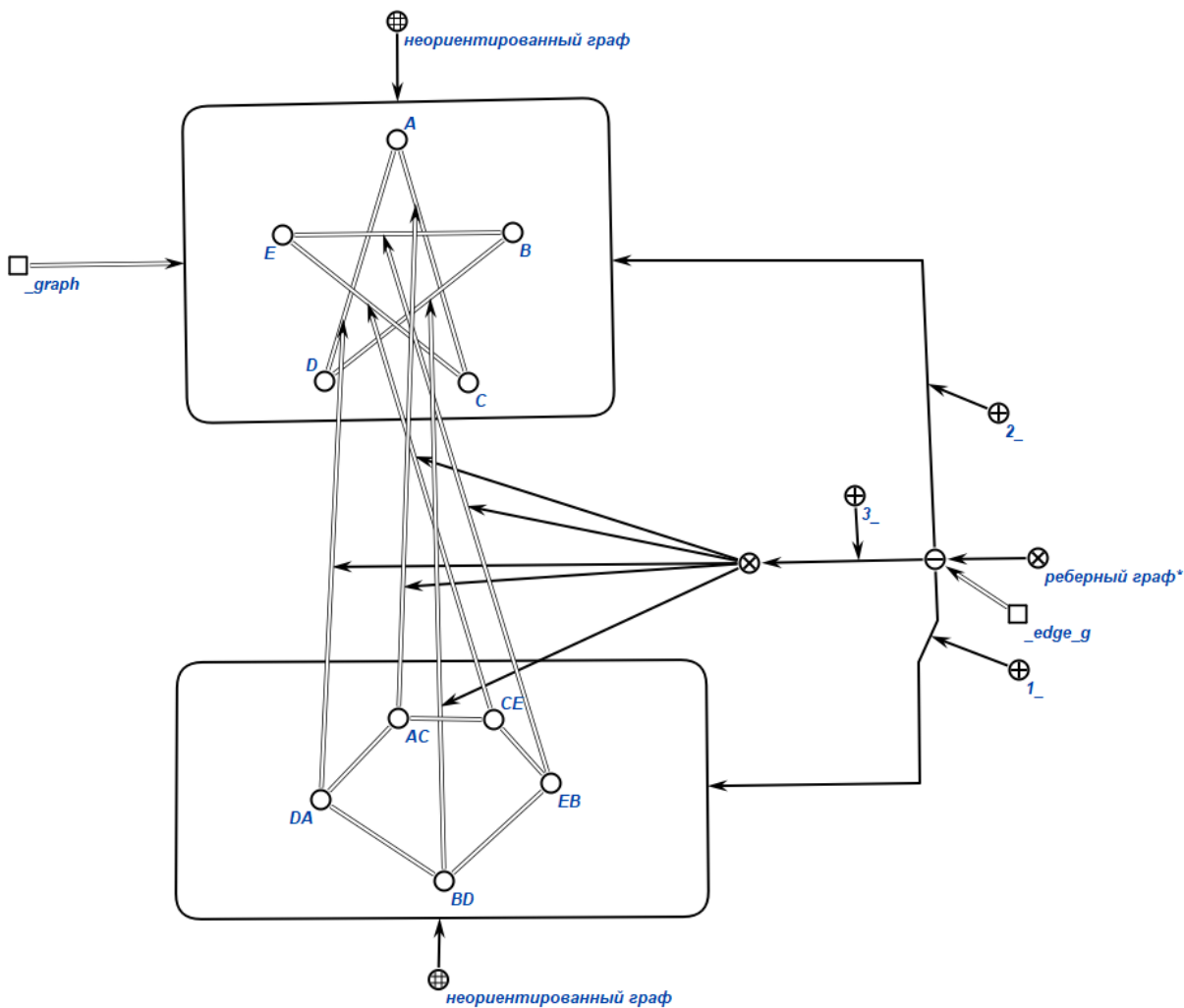
Вершина C. Смежные ей: A и E. В реберном графе создаем ребро между вершинами AC и EC. Удаляем C из _not_checked_vertices.

Вершина D. Смежные ей: A и B. В реберном графе создаем ребро между вершинами AD и DB. Удаляем D из _not_checked_vertices.

Вершина E. Смежные ей: C и B. В реберном графе создаем ребро между вершинами EC и EB. Удаляем B из _not_checked_vertices.

_not_checked_vertices стало пусто. Значит, мы проверили все вершины и создали связи в реберном графе. Можем удалить множество непроверенных вершин, переменную текущей проверяемой вершины и получить ответ.

6. Результат работы алгоритма



3 Список литературы

OSTIS GT [В Интернете] // База знаний по теории графов OSTIS GT. - 2011 г.. - http://ostisgraphstheo.sourceforge.net/index.php/Заглавная_страница.

Харарри Ф. Теория графов [Книга]. - Москва : Едиториал УРСС, 2003.