

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 рёберный граф для данного неориентированного графа.

Выполнил:

Е. М. Римонт

Студент группы
321702

Проверил:

Н. В. Малиновская

Минск 2024

Содержание

1	Введение	2
2	Список понятий	2
3	Тестовые примеры	4
3.1	Тест 1:	4
3.2	Тест 2	5
3.3	Тест 3	6
3.4	Тест 4	7
4	Пример работы алгоритма в семантической памяти	8
4.1	Краткое описание:	8
4.2	Демонстрация на тесте 5:	8
5	Заключение	17

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. **Граф** - совокупность непустого множества вершин и множества пар вершин (рёбер)
2. **Неориентированный граф** (абсолютное понятие)-граф, в котором все рёбра являются звеньями, то есть порядок двух концов ребра графа не существует
 - (a) Вершина (относительное понятие, ролевое отношение);
 - (b) Связка (относительное понятие, ролевое отношение).

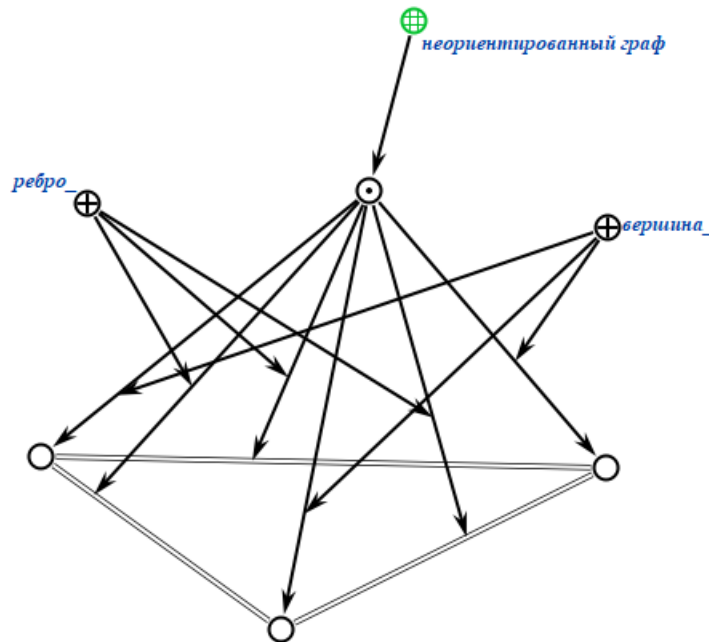


Рис. 1: Абсолютное понятие неориентированного графа.

3. **Реберный граф** - такой граф $L(G)$ для данного графа G , который удовлетворяет следующим условиям:
 - (a) Любая вершина графа $L(G)$, представляет собой ребро графа G
 - (b) Две вершины графа $L(G)$ смежны тогда и только тогда, когда их соответствующие рёбра смежны в графе G

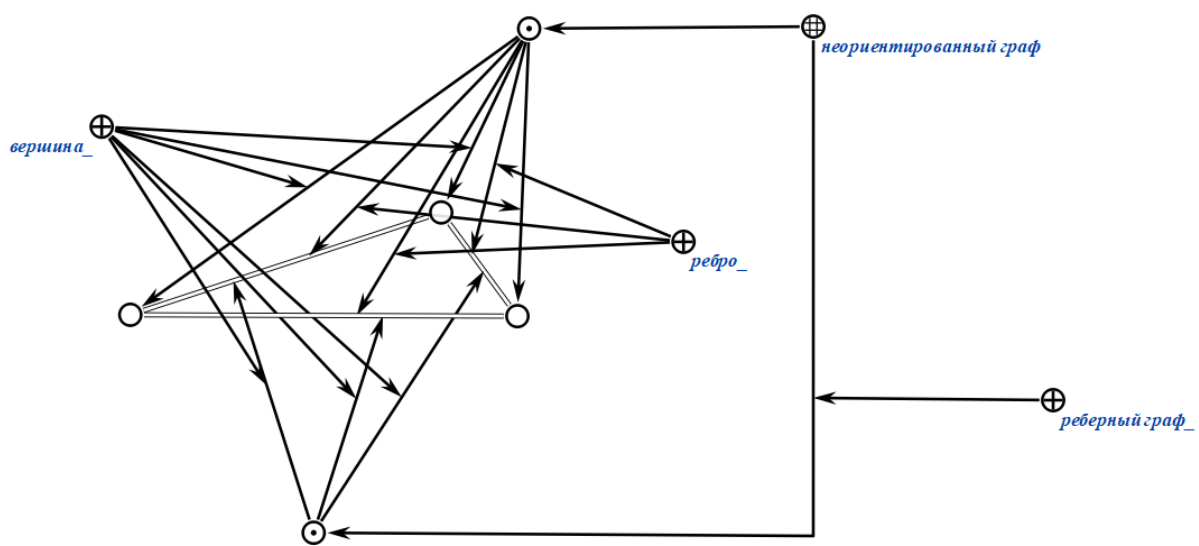


Рис. 2: Понятие реберного графа.

3 Тестовые примеры

Во всех тестах графы будут приведены в сокращенной форме.

3.1 Тест 1:

Вход:

Необходимо найти реберный граф для данного неориентированного графа

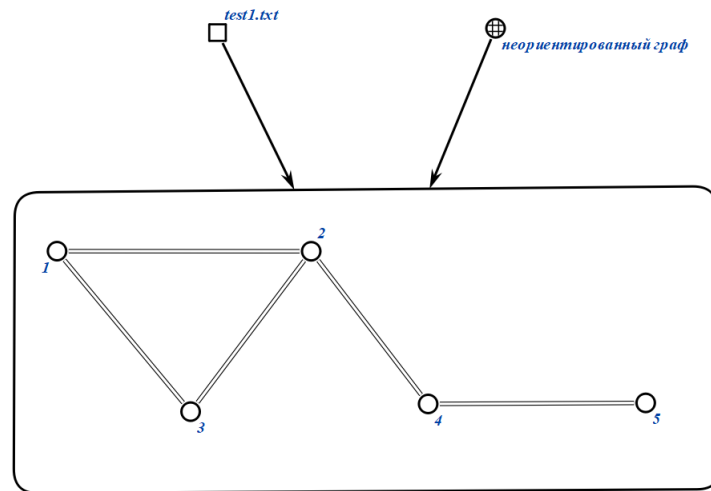


Рис. 3: Вход теста 1.

Выход: Будет найден реберный граф:

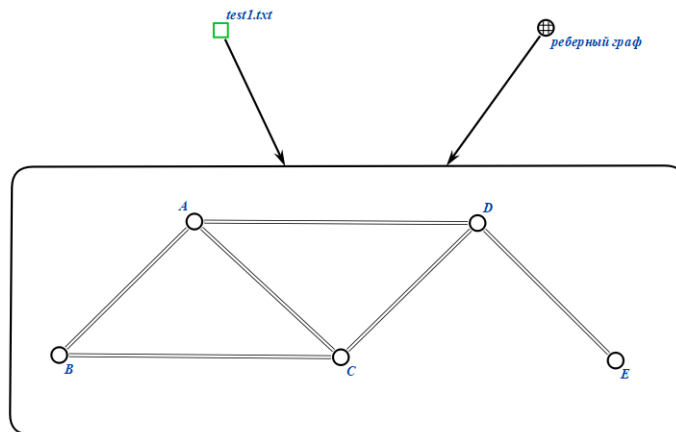


Рис. 4: Выход теста 1.

3.2 Тест 2

Вход: Необходимо найти реберный граф для данного неориентированного графа

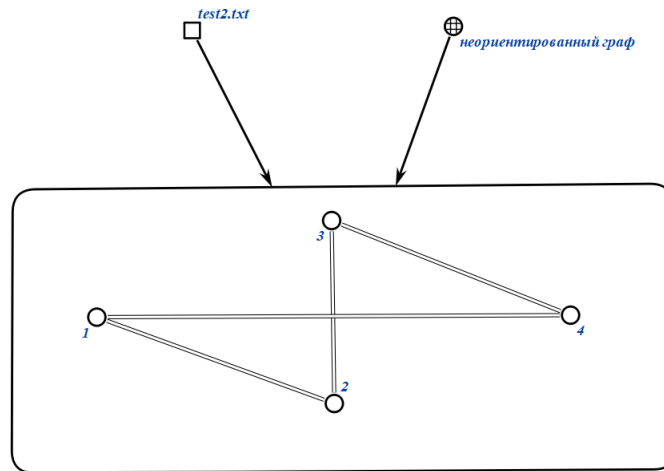


Рис. 5: Вход теста 2.

Выход: Будет найден реберный граф:

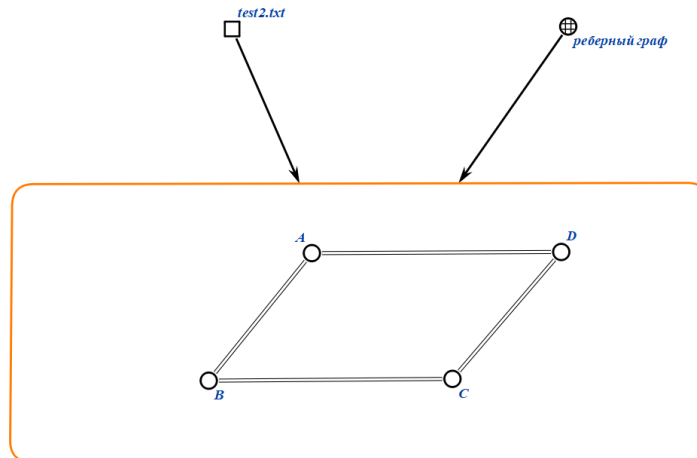


Рис. 6: Выход теста 2.

3.3 Тест 3

Вход: Необходимо найти реберный граф для данного неориентированного графа

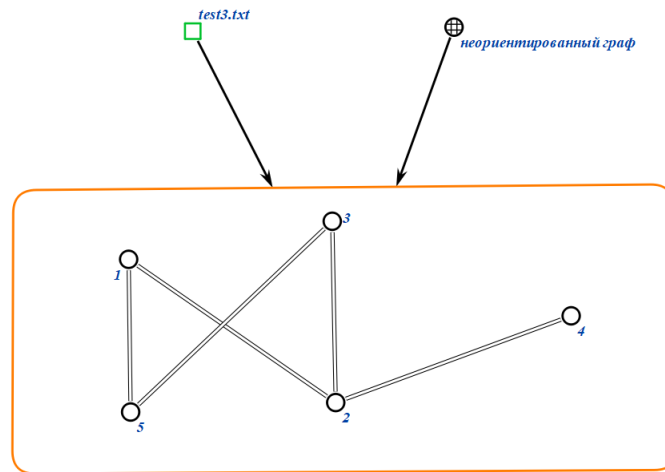


Рис. 7: Вход теста 3.

Выход: Будет найден реберный граф:.

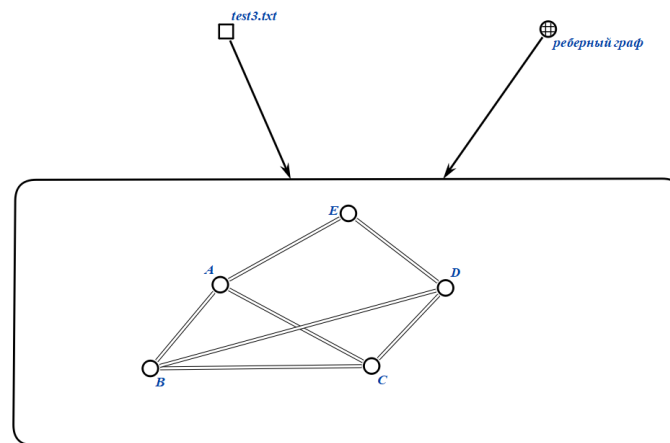


Рис. 8: Выход теста 3.

3.4 Тест 4

Вход: Необходимо найти реберный граф для данного неориентированного графа

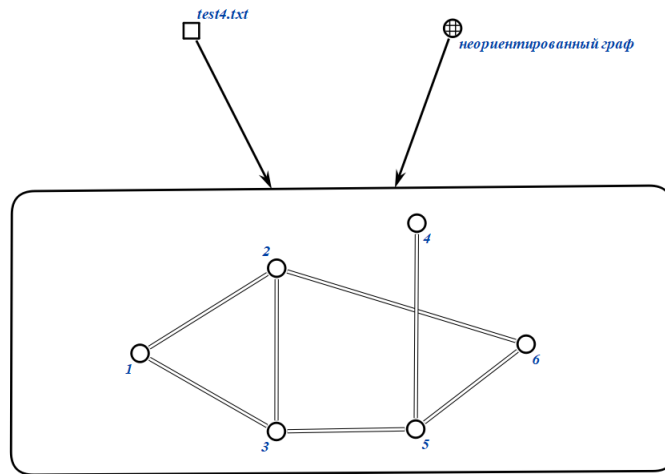


Рис. 9: Вход теста 4.

Выход: Будет найден реберный граф:

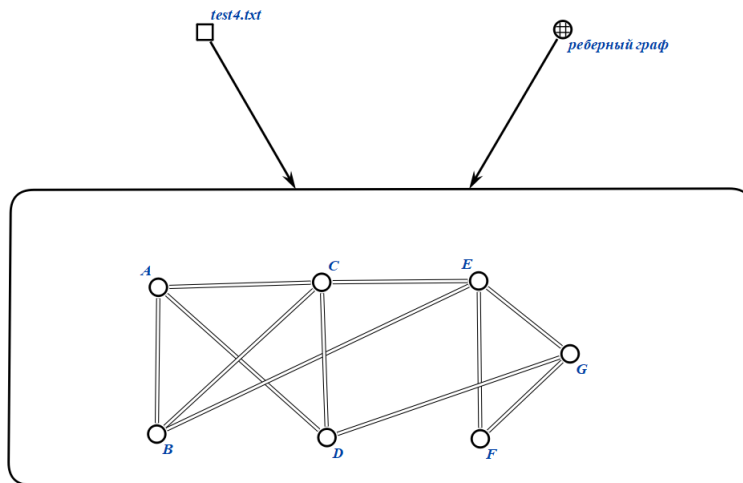


Рис. 10: Выход теста 4.

4 Пример работы алгоритма в семантической памяти

4.1 Краткое описание:

1. Создаем список пар, который будет хранить информацию о смежных ребрах
2. Формируем новую волну и добавляем в нее первое ребро
3. Добавляем в текущую волну рёбра, которые смежны уже добавленному в волну ребру
4. Список пар принимает пары ребер, которые смежны, где первое ребро - первое добавленное в волну ребро, второе ребро - каждое из оставшихся добавленных в волну. Таким образом список пар может принять не одну пару смежных ребер
5. Формируем новую волну для следующего выбранного ребра, добавляем в него выбранное ребро. Добавляем в волну далее все смежные первому добавленному ребру рёбра. Список пар принимает пары смежных ребёр аналогично пункту 4. Повторяем пункт 5 до обозревания всех ребер.
6. Если пройдены все ребра, переходим к построению рёберного графа
7. Пары найденных ребер входного неориентированного графа становятся парами смежных вершин, т.е. рёбрами выходного рёберного графа

4.2 Демонстрация на тесте 5:

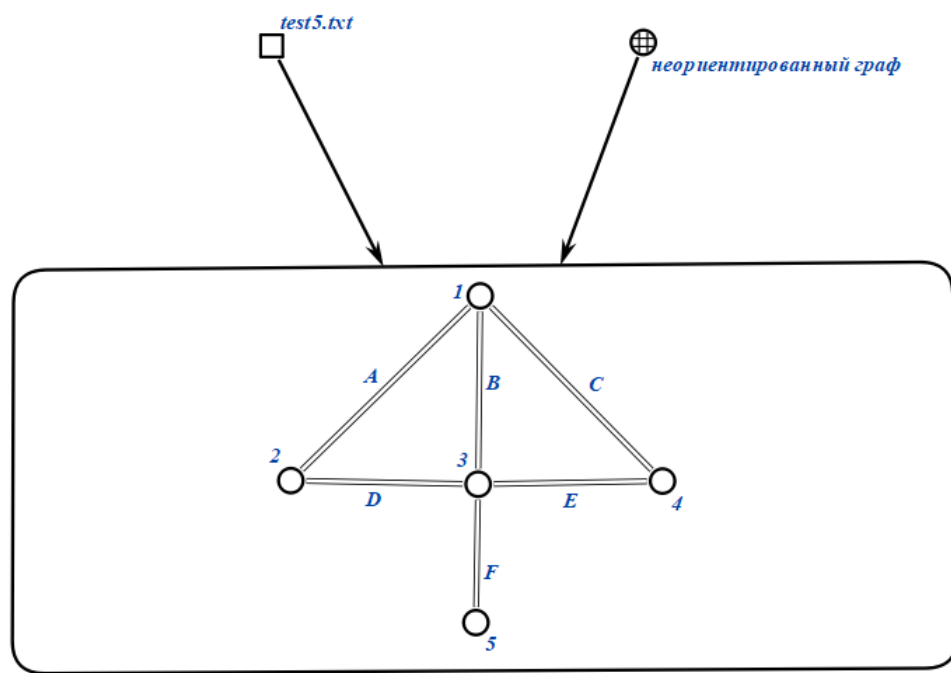


Рис. 11: Вход теста 5.

1. *adjacent_matrix* получит в качестве значения ss-узел неориентированного графа;

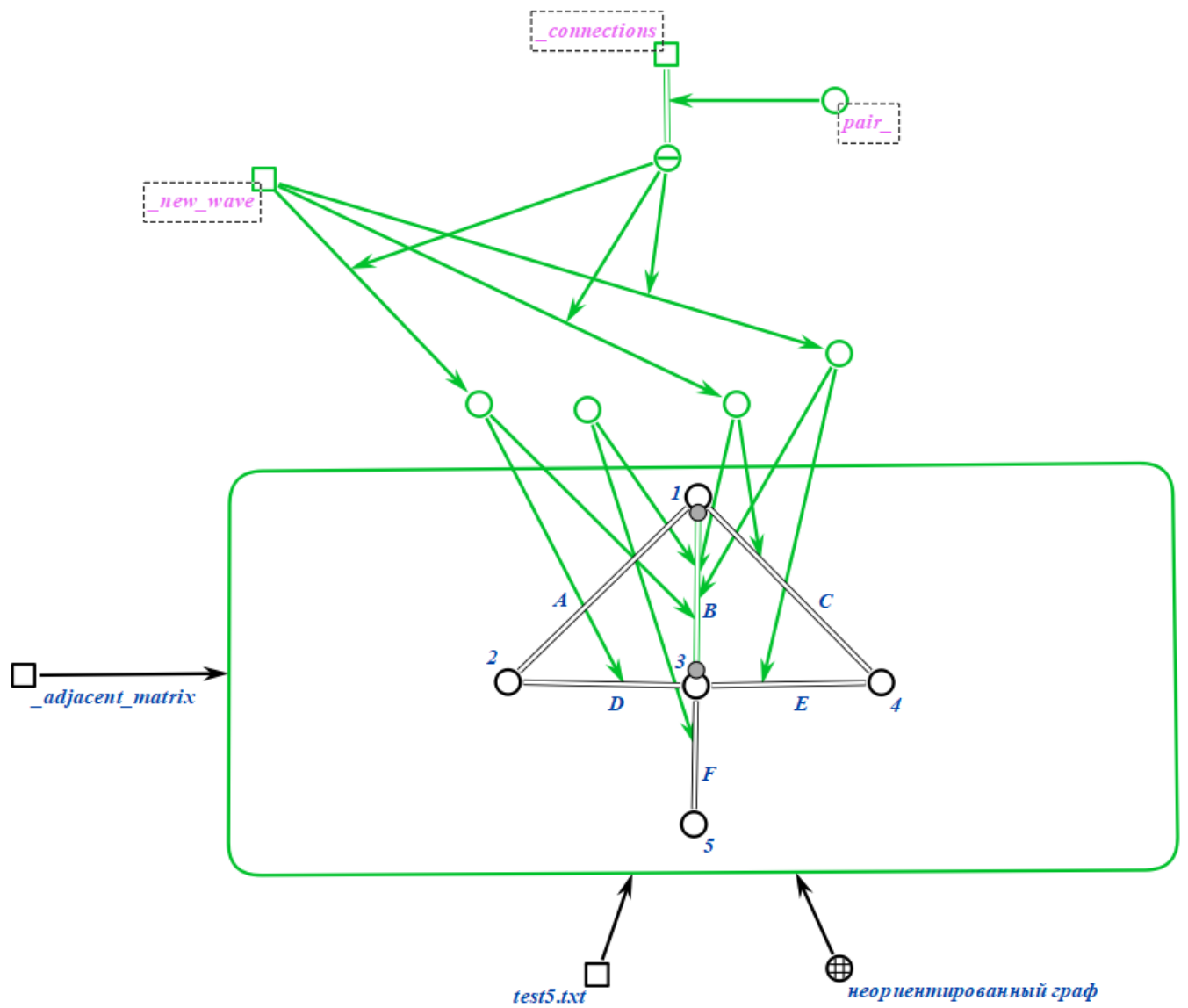


Рис. 14: Действие 3.

4. На этом этапе создается новая волна, которая сперва имеет лишь следующее выбранное ребро *B*. После в созданную волну добавляются смежные этому ребру другие ребра *C, D, E, F*, которые сразу добавляются в список *connections*

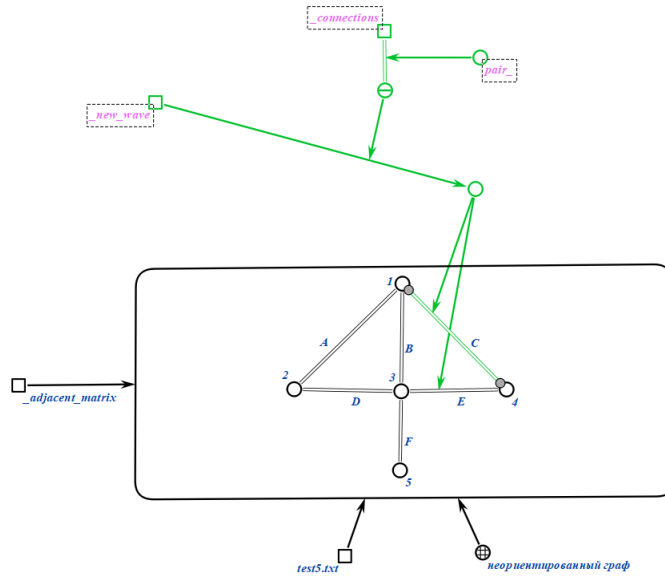


Рис. 15: Действие 4.

5. Для следующего выбранного ребра **C** найдено всего одно смежное ребро **E**, пары с которым ещё нет в списке пар **connections**

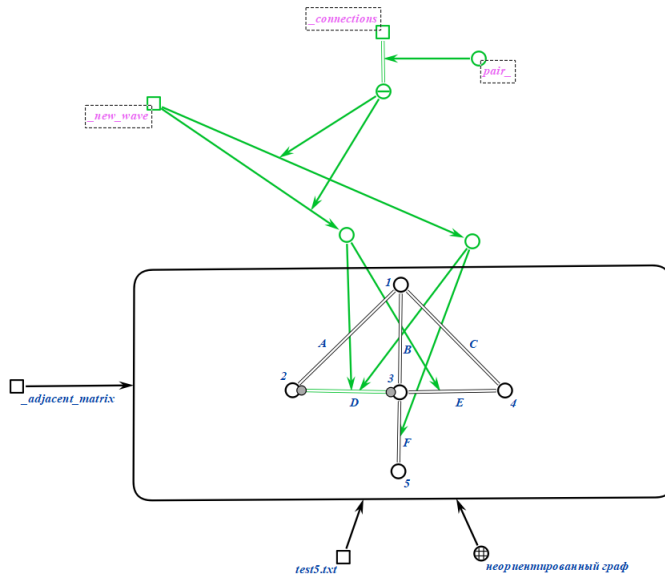


Рис. 16: Действие 5.

6. Для следующего выбранного ребра **D**, найдено два ребра **E, F**, пары которых вместе с **D** заносятся в список пар **connections**

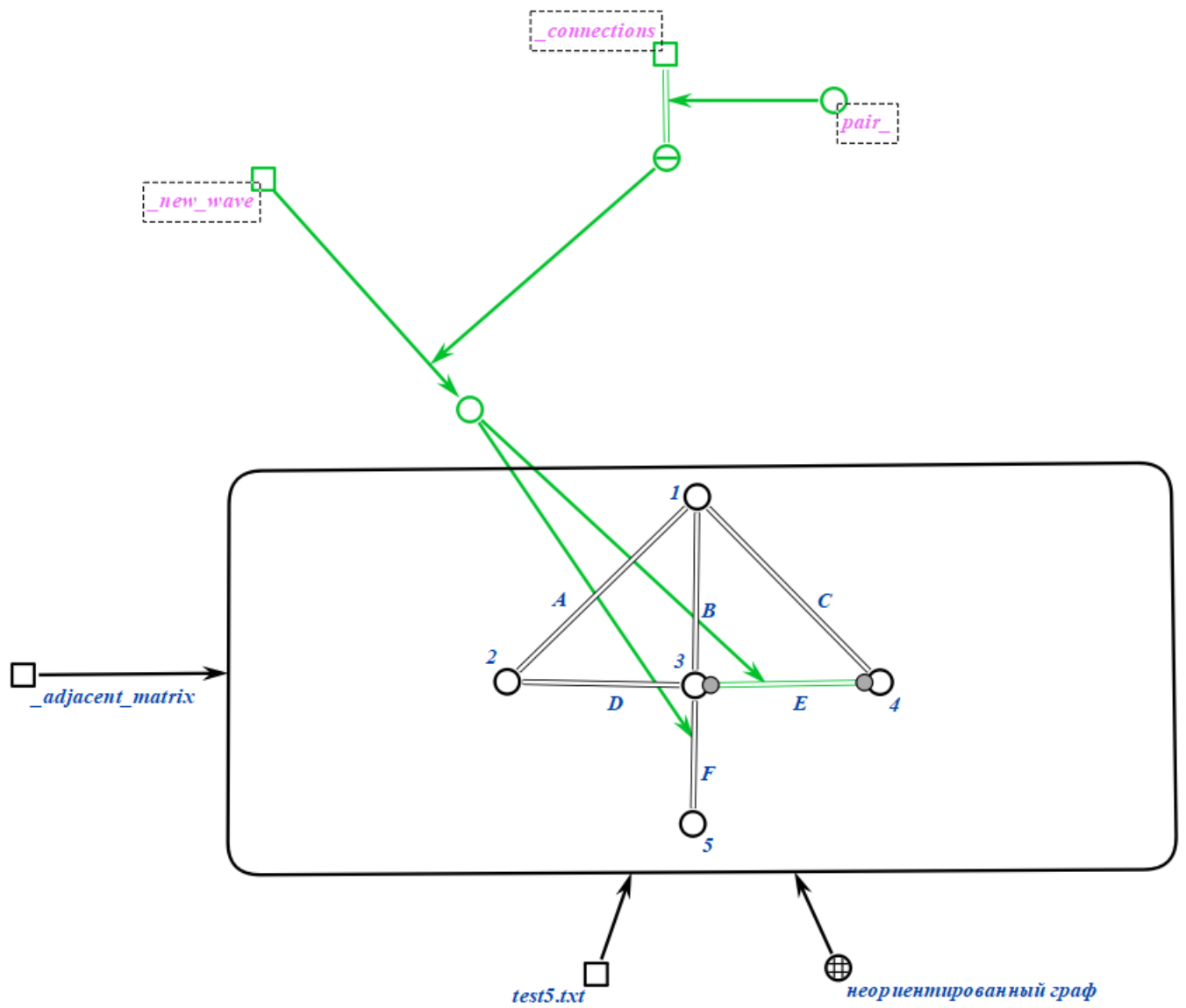


Рис. 17: Действие 6.

7. Для ребра E найдено всего одно смежное ребро F , пары с которым ещё нет в списке пар *connections*

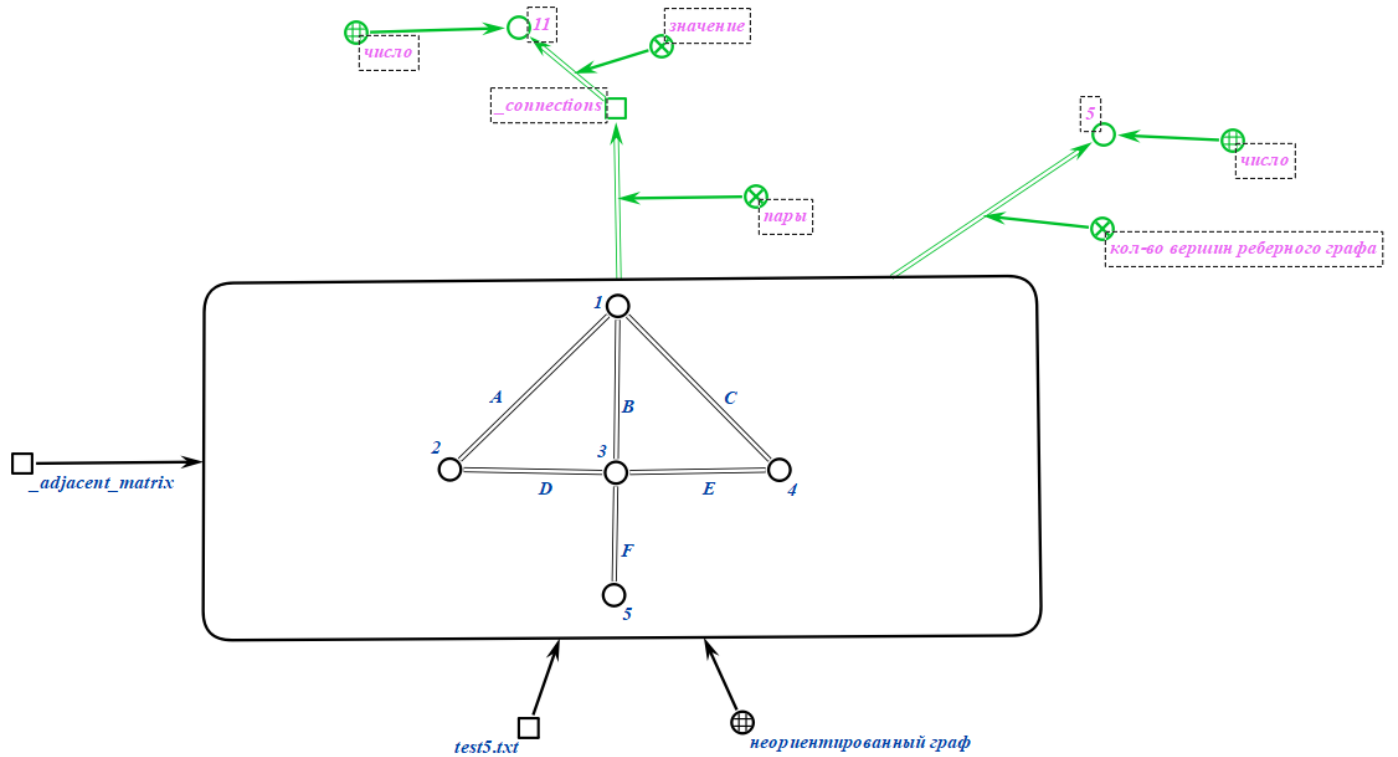


Рис. 18: Действие 7.

8. Для ребра **F** новая волна не создается, т.к. оно было записано во всех возможных парах смежных ребер при прохождении предыдущих ребёр, а значит рассматривать ребро **F** не имеет смысла. Все ребра были осмотрены.

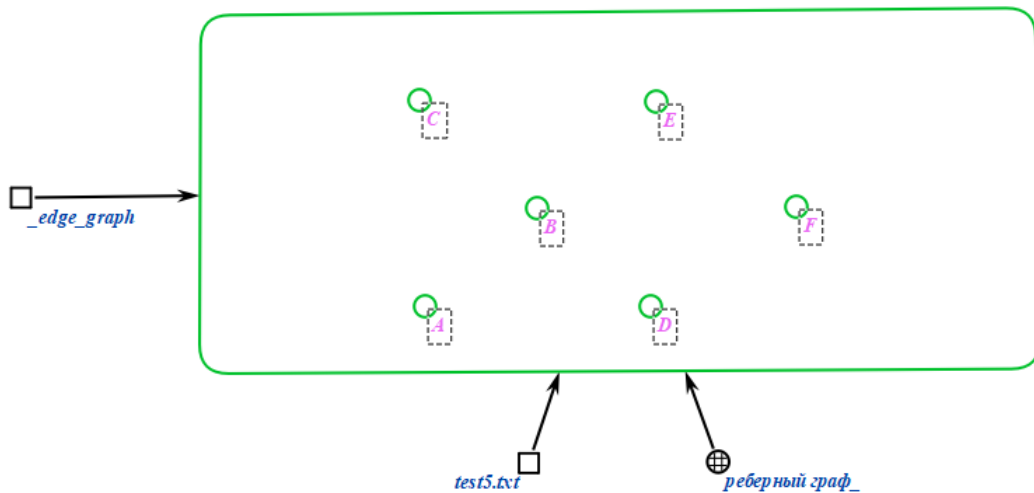


Рис. 19: Действие 8.

9. Список **connections** получил 11 пар, значит конечный реберный граф **edge_graph** будет иметь 11 рёбер. Начальный граф имеет количество рёбер равное 6, значит конечный реберный граф будет иметь 6 вершин.

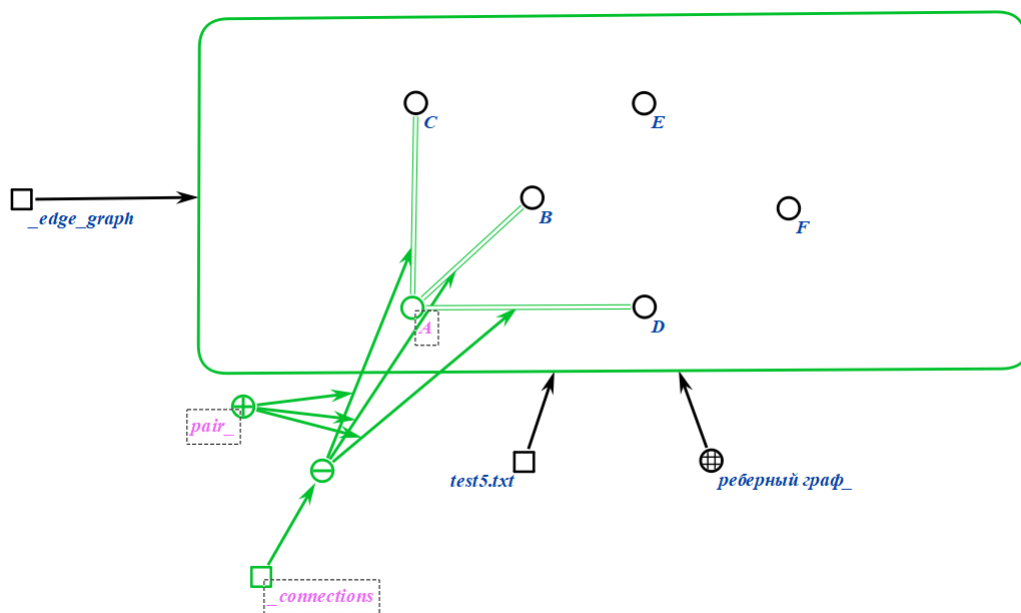


Рис. 20: Действие 9.

10. Ребёрный граф *edge_graph* принимает ребра для вершины *A* в виде пар вершин из списка *connections*: *A-B*, *A-C*, *A-D*

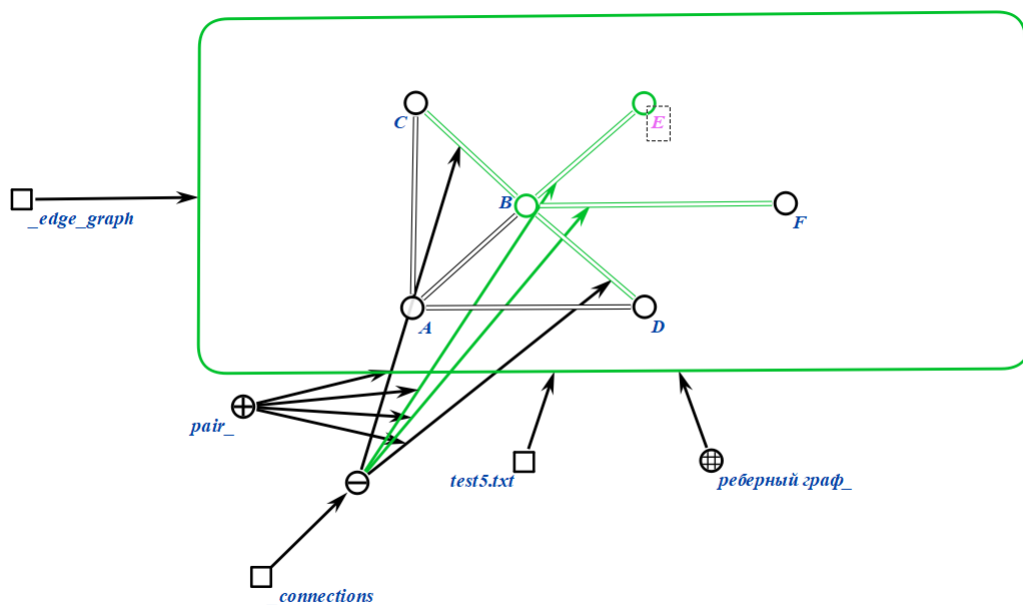


Рис. 21: Действие 10.

11. Ребёрный граф *edge_graph* принимает ребра для вершины *B* в виде пар вершин из списка *connections*: *B-C*, *B-D*, *B-E*, *B-F*

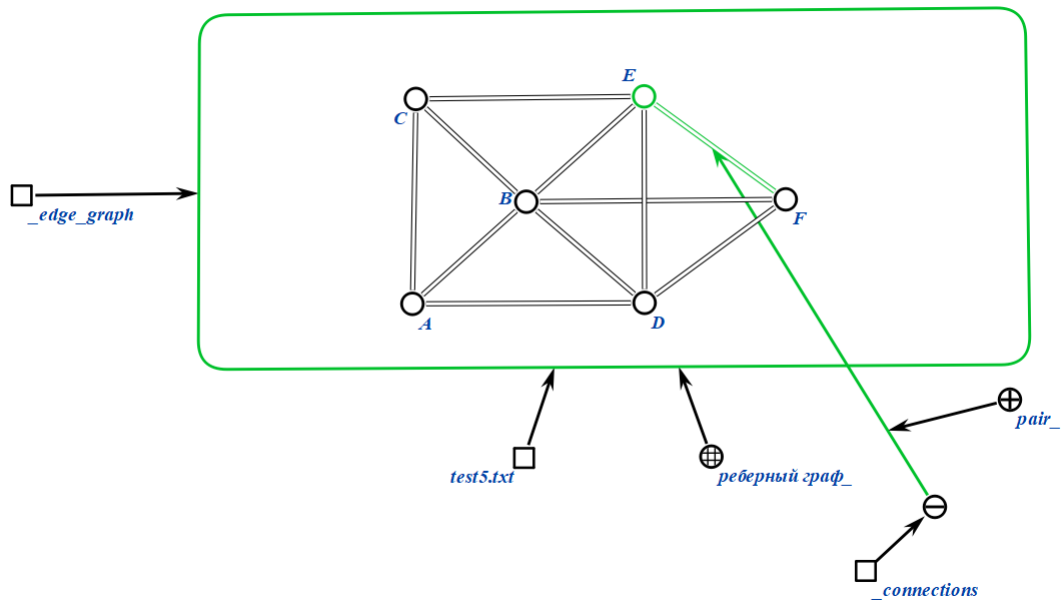


Рис. 24: Действие 13.

14. Ребёрный граф *edge_graph* принимает ребра для вершины *E* в виде пар вершин из списка *connections*: *E-F*

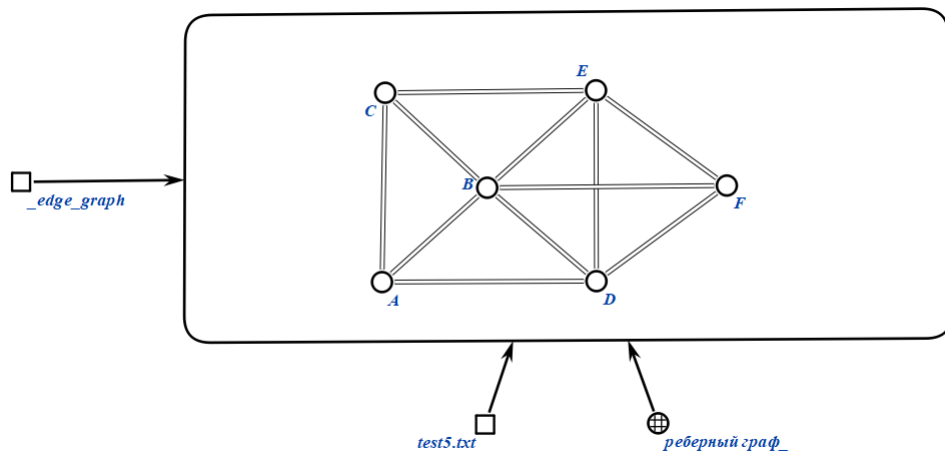


Рис. 25: Выход теста 5.

15. Все пары из списка *connections* были получены графом *edge_graph* Ребёрный граф *edge_graph* построен.

5 Заключение

В заключении у нас получилось формализовать поставленную задачу. Мы нашли нужный нам реберный граф для неориентированного графа. Реализовали алгоритм его нахождения, который работает на любом неориентированном графе.