

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

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

# 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
  - (а) Вершина (относительное понятие, ролевое отношение);
  - (б) Связка (относительное понятие, ролевое отношение).

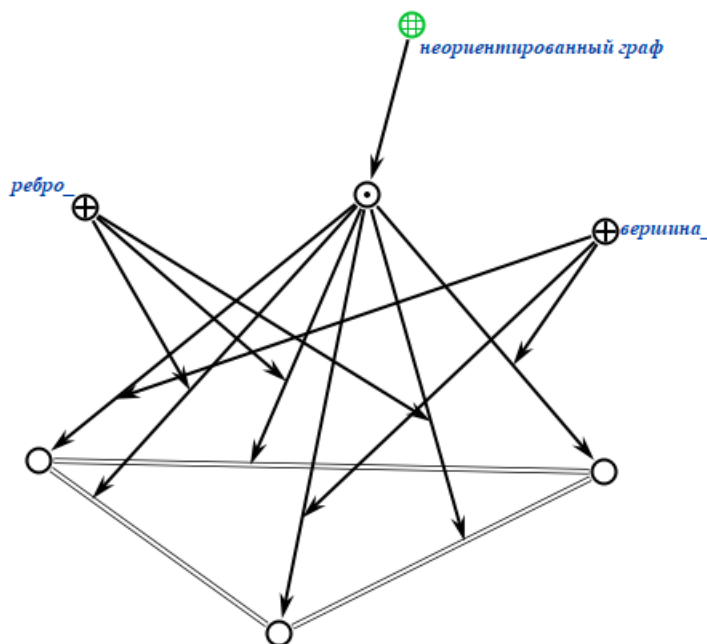


Рис. 1: Абсолютное понятие неориентированного графа

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

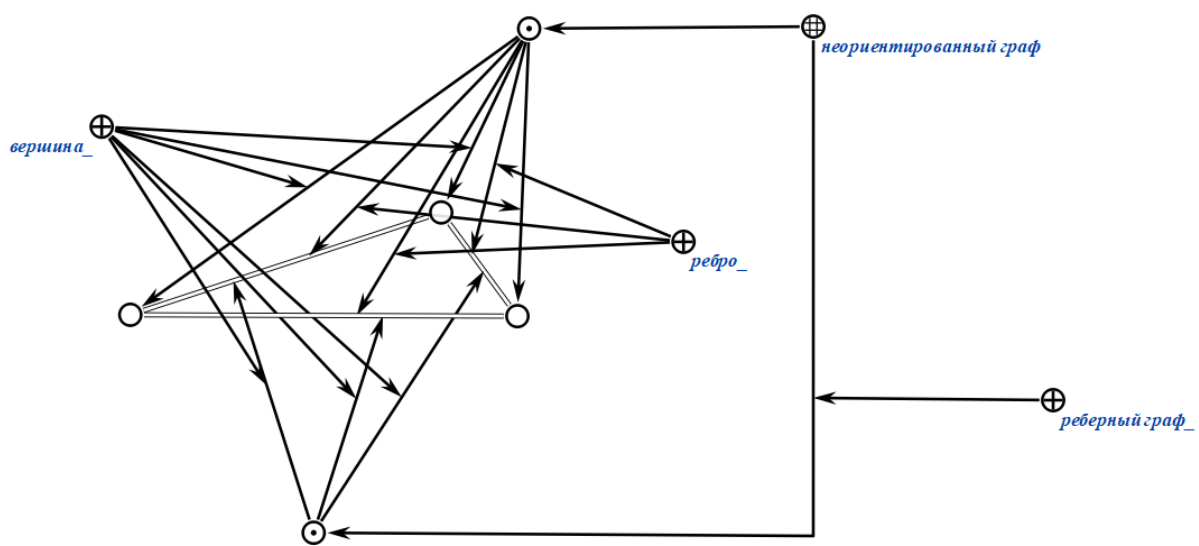


Рис. 2: Понятие реберного графа

### 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

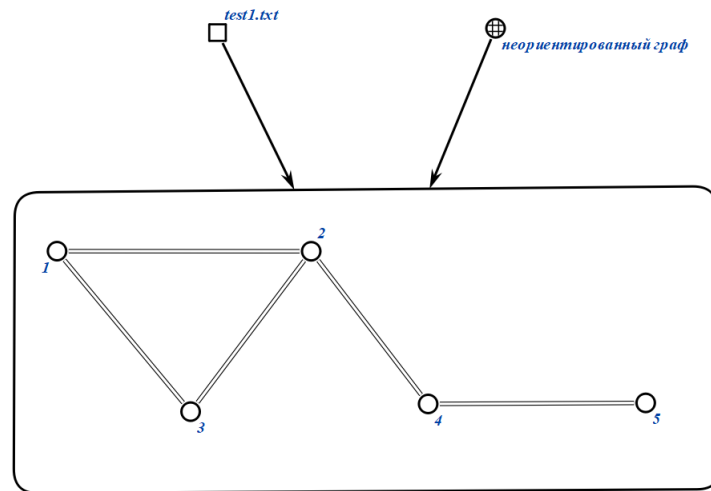


Рис. 3: Вход теста 1

**Выход:** Будет найден реберный граф:

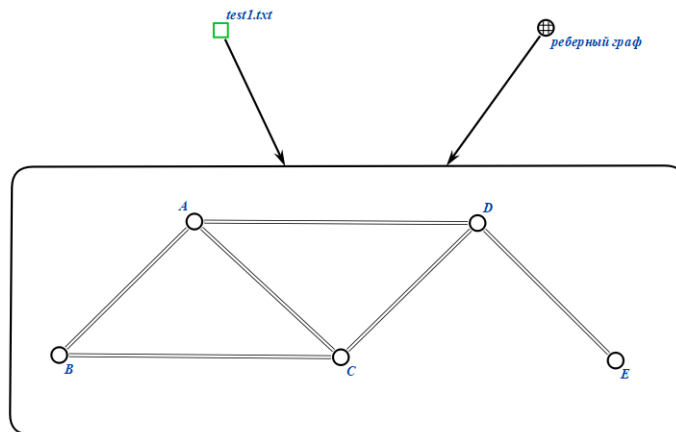


Рис. 4: Выход теста 1

### 3.2 Тест 2

**Вход:** Необходимо найти реберный граф для данного неориентированного графа

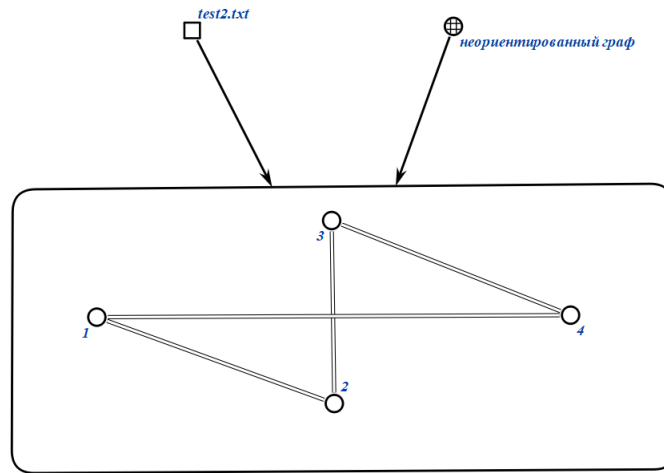


Рис. 5: Вход теста 2

**Выход:** Будет найден реберный граф:

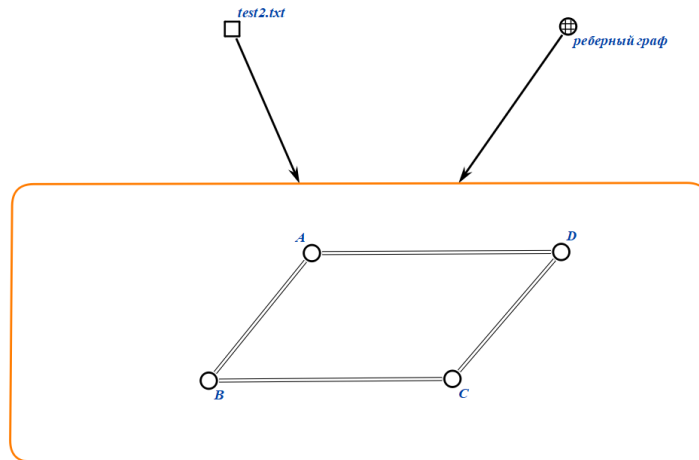


Рис. 6: Выход теста 2

### 3.3 Тест 3

**Вход:** Необходимо найти реберный граф для данного неориентированного графа

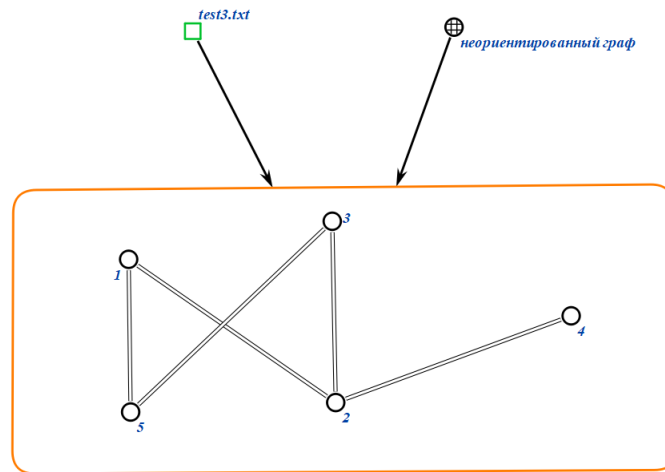


Рис. 7: Вход теста 3

**Выход:** Будет найден реберный граф:.

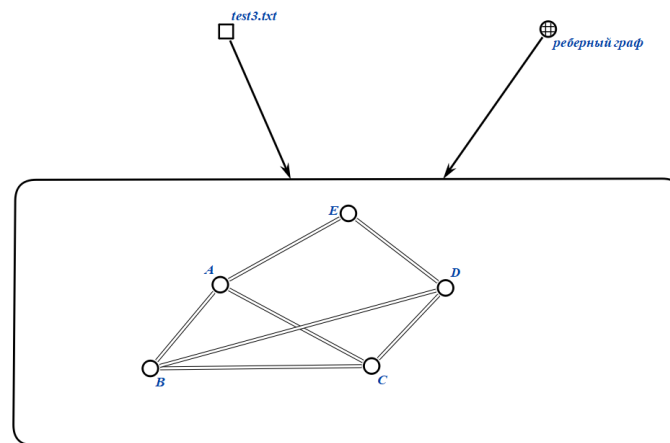


Рис. 8: Выход теста 3

### 3.4 Тест 4

**Вход:** Необходимо найти реберный граф для данного неориентированного графа

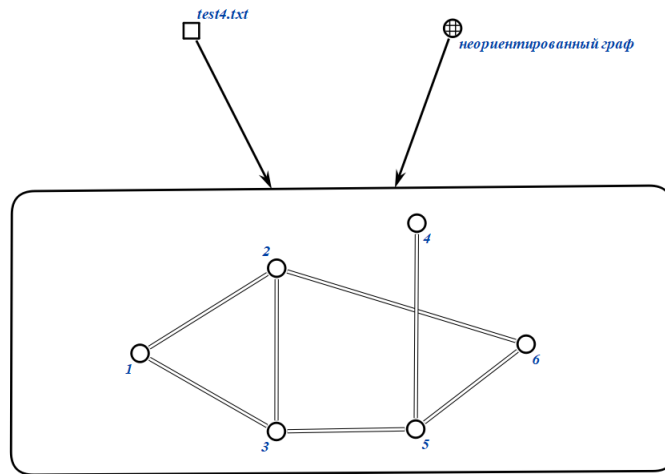


Рис. 9: Вход теста 4

**Выход:** Будет найден реберный граф:

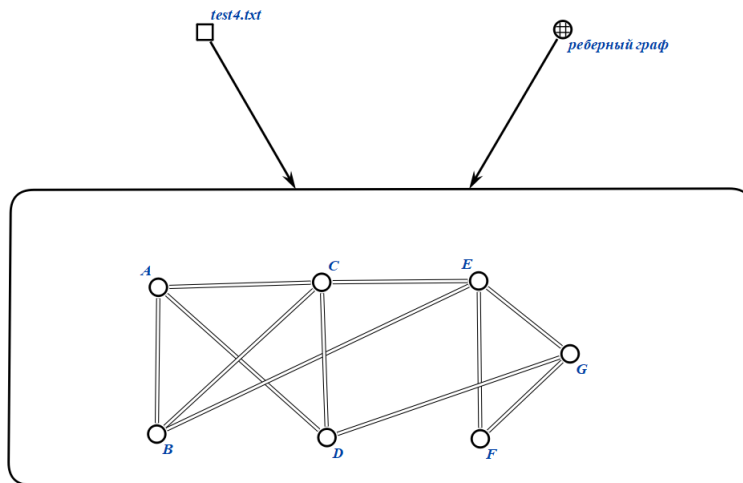


Рис. 10: Выход теста 4



## 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:

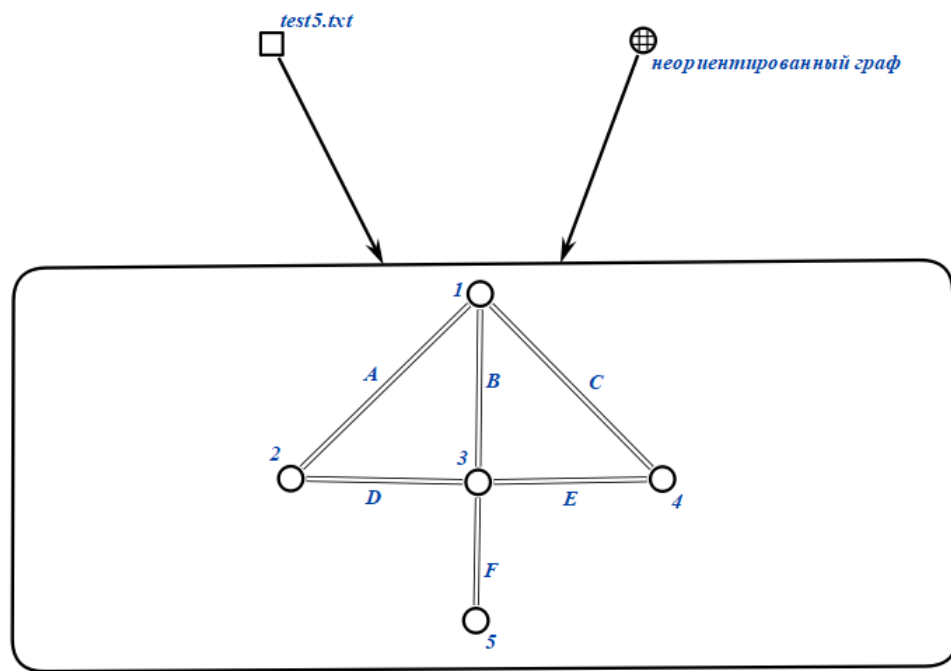


Рис. 11: Вход теста 5

1. *adjacent\_matrix* получит в качестве значения sc-узел неориентированного графа;

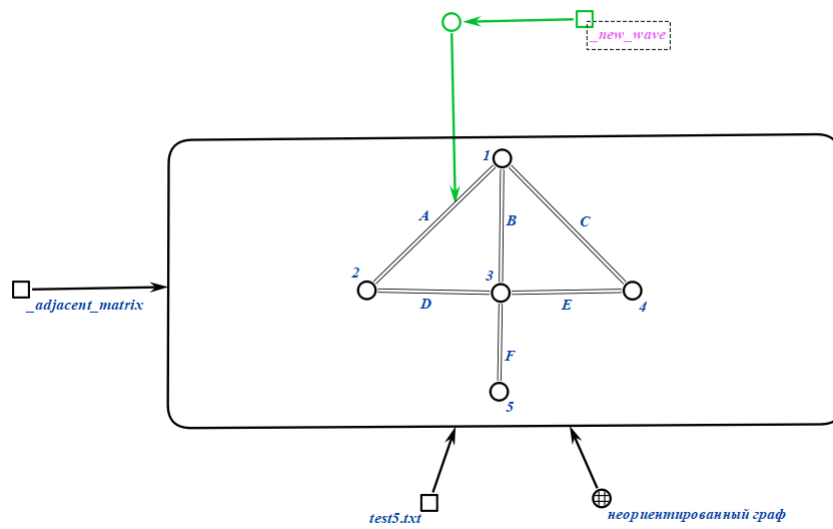


Рис. 12: Действие 1

2. Создастся волна, которая сразу включает в себя первое ребро.

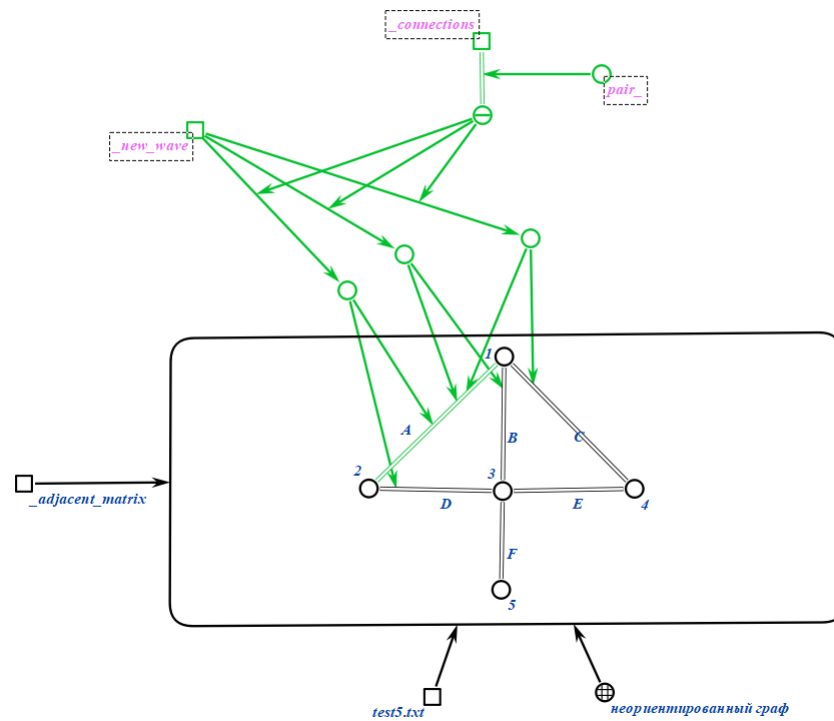


Рис. 13: Действие 2

3. На этом этапе в волну добавится 3 пары смежных ребер, список *connections* получит эти значения в качестве 3-х различных пар вершин.

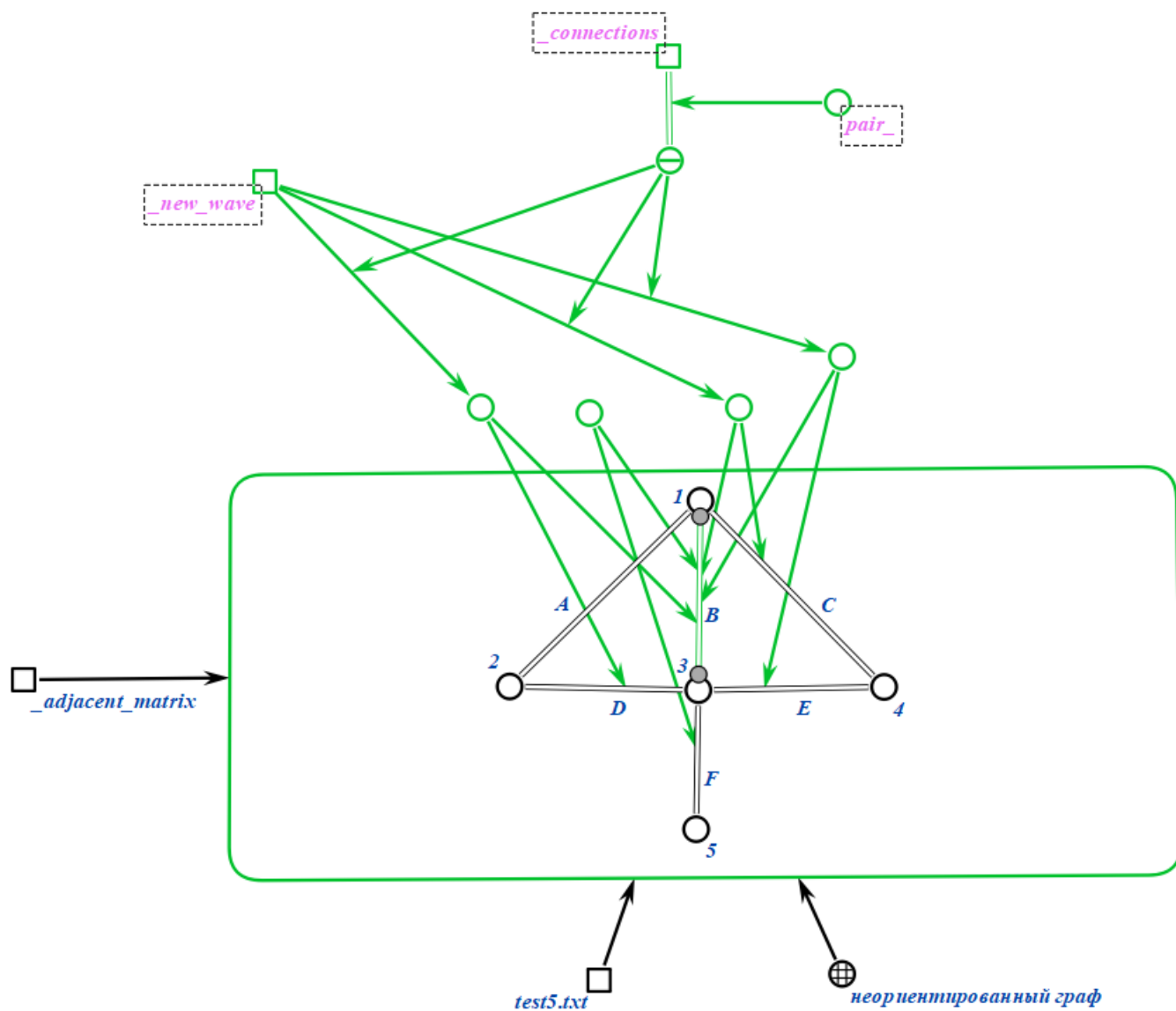


Рис. 14: Действие 3

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

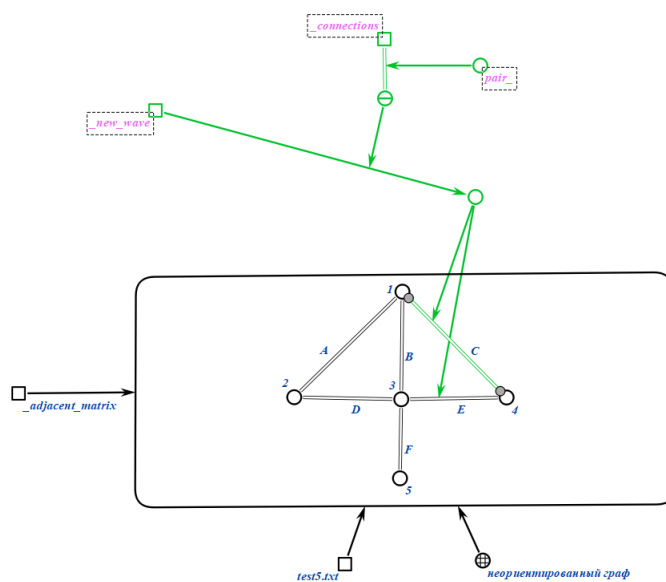


Рис. 15: Действие 4

5. Алгоритм повторяется для следующего ребра

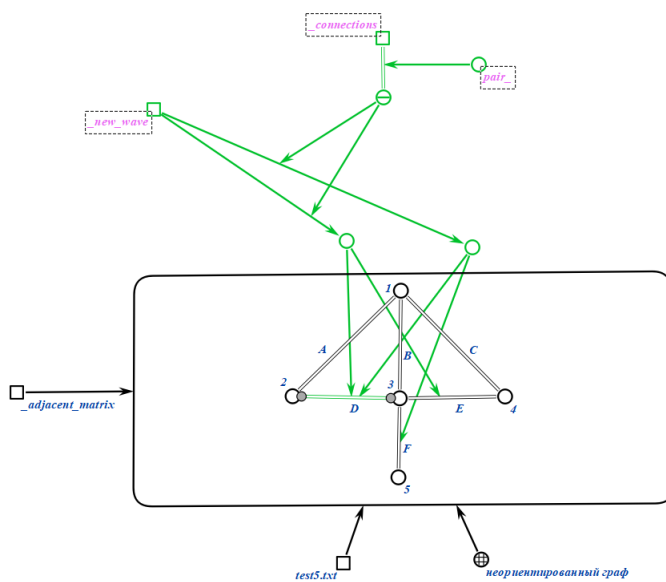
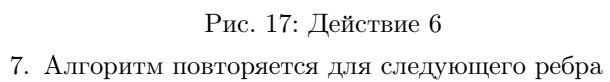


Рис. 16: Действие 5

6. Алгоритм повторяется для следующего ребра



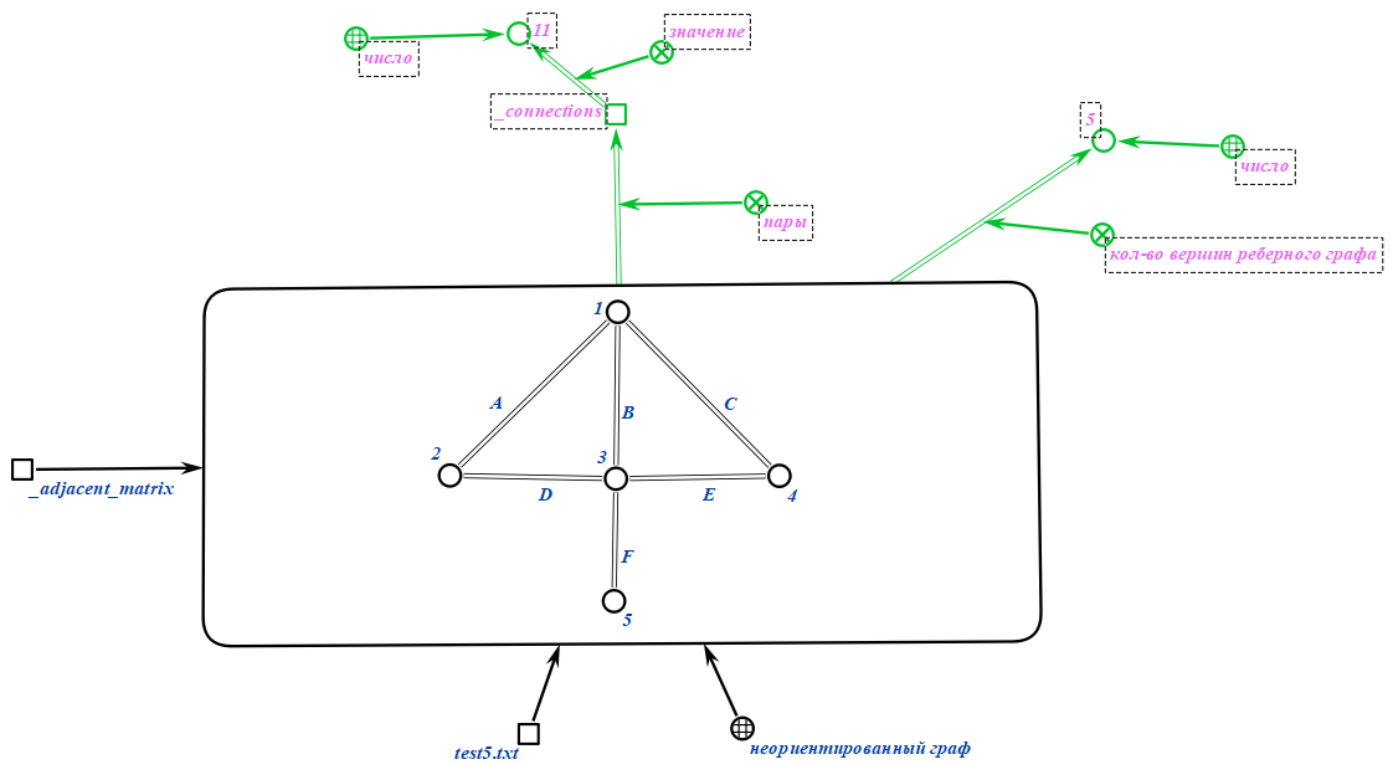


Рис. 18: Действие 7

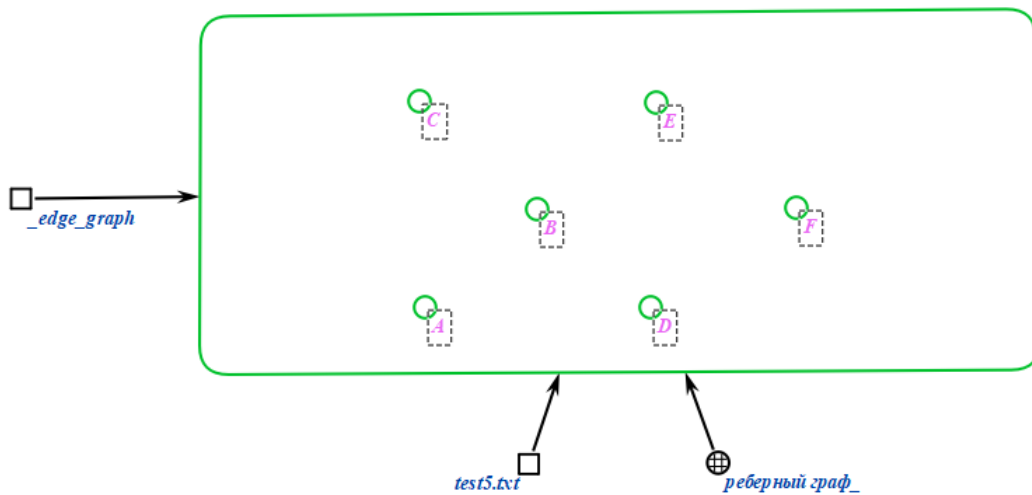


Рис. 19: Действие 8

8. Все ребра были рассмотрены. Список **connections** получил 11 пар, значит конечный реберный граф **edge\_graph** будет иметь 11 рёбер. Начальный же граф имеет количество рёбер равно 6, значит конечный рёберный граф будет иметь 6 вершин.

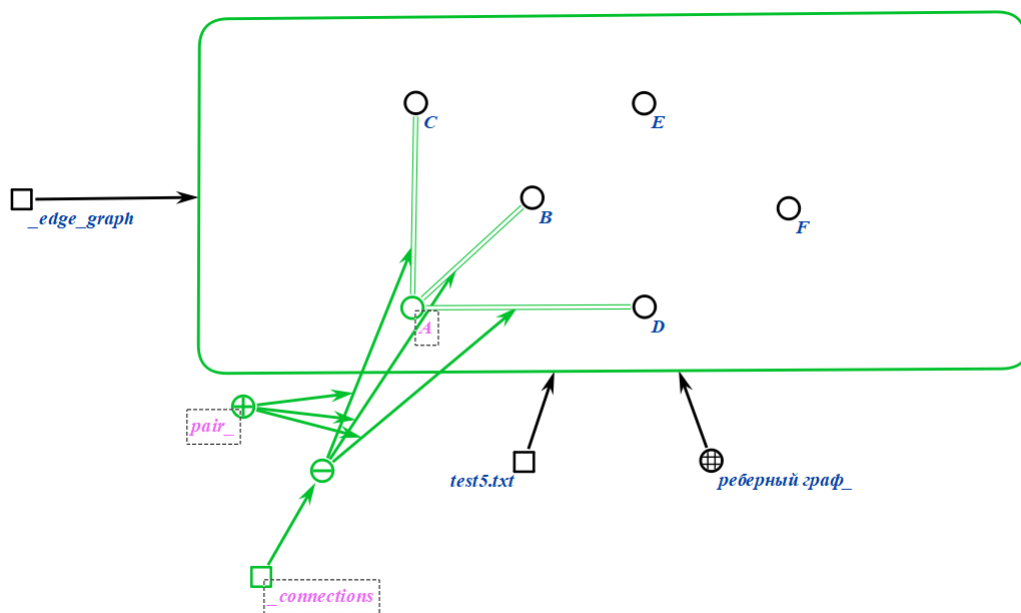


Рис. 20: Действие 9

9. Ребёрный граф *edge\_graph* принимает ребра в виде пар вершин из списка *connections*

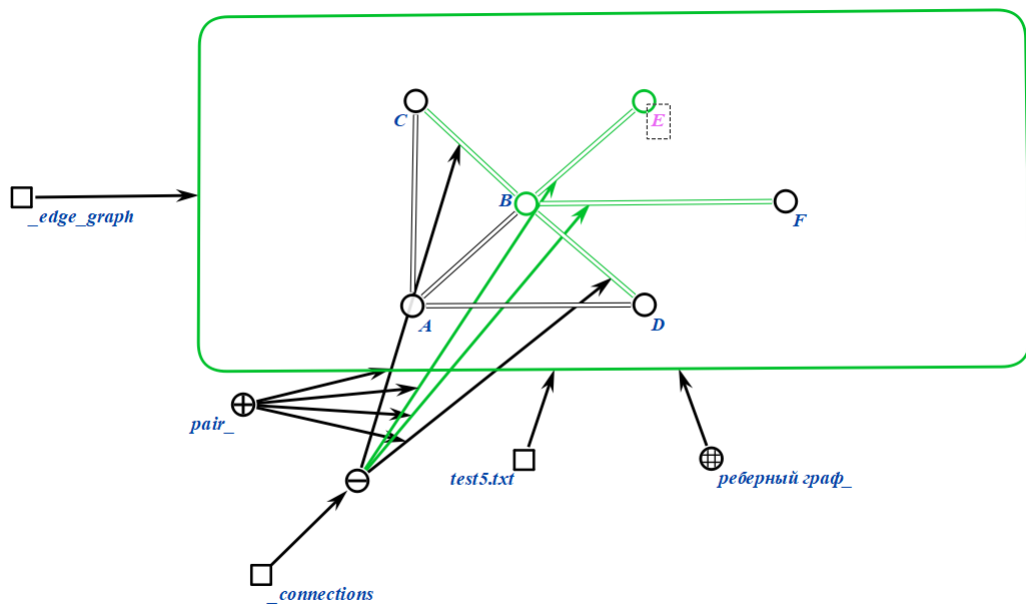
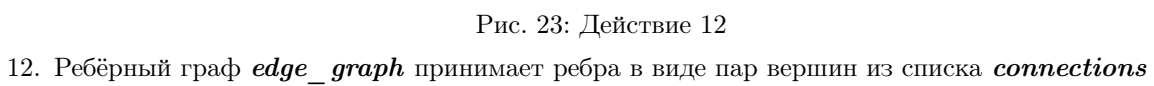
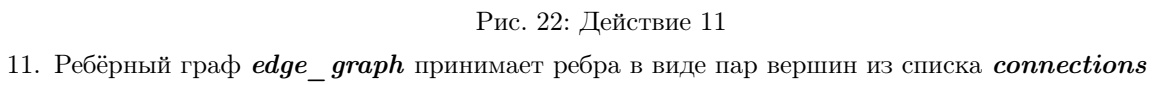


Рис. 21: Действие 10

10. Ребёрный граф *edge\_graph* принимает ребра в виде пар вершин из списка *connections*





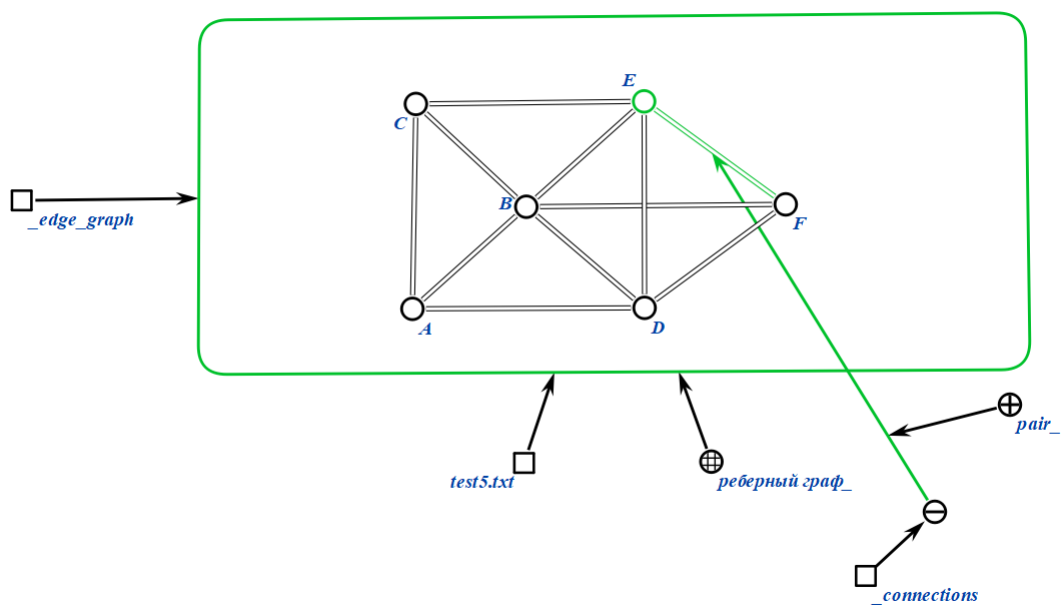


Рис. 24: Действие 13

13. Ребёрный граф *edge\_graph* принимает ребра в виде пар вершин из списка *connections*

## 5 Заключение

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