

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 данных в интеллектуальных
системах»

на тему

**Задача нахождения дерева кратчайших путей в
неориентированном взвешенном графе**

Выполнил:

А. С. Балабенко

Студент группы
421701

Проверил:

Н. В. Малиновская

Минск 2025

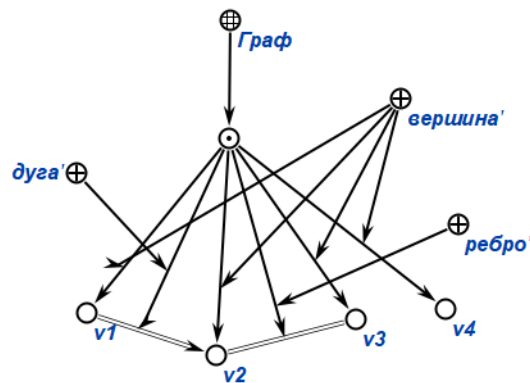
1 ВВЕДЕНИЕ

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

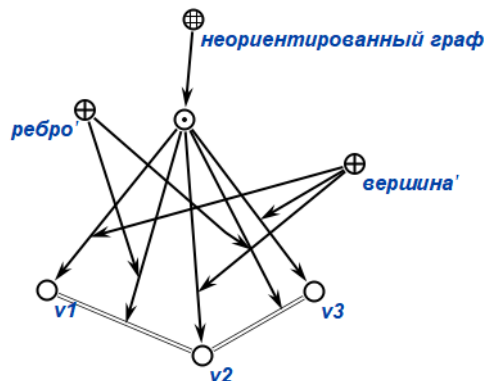
Задача: Найти дерево кратчайших путей в неориентированном взвешенном графе.

2 СПИСОК ПОНЯТИЙ

1. **Граф** - математическая абстракция реальной системы любой природы, объекты которой обладают парными связями.
(совокупность точек, соединенных линиями. Точки называются вершинами, или узлами, а линии – ребрами, или дугами.) :



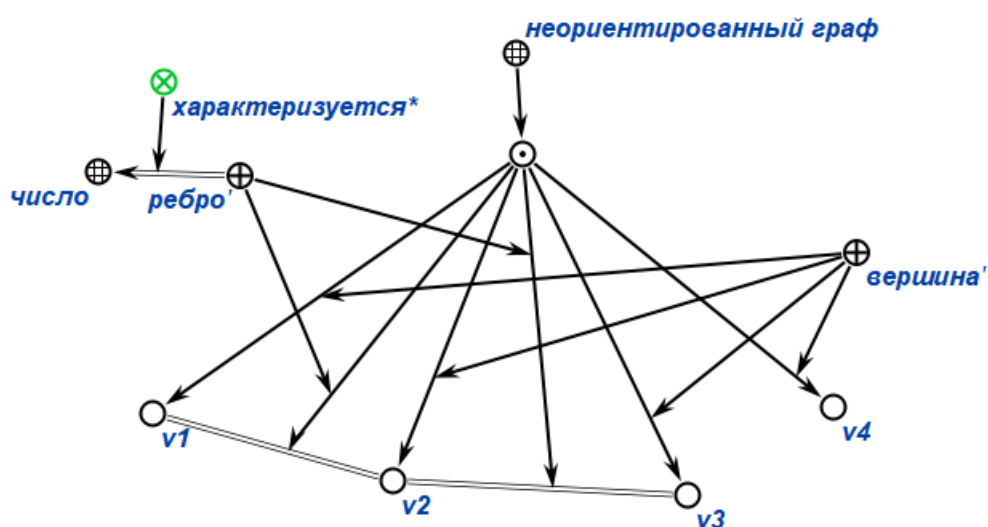
2. **Неориентированный граф** — Граф, ни одному ребру которого не присвоено направление.



3. **Матрица инцидентности графа** — это матрица, количество строк в которой соответствует числу вершин, а количество столбцов – числу рёбер. В ней указываются связи между

инцидентными элементами графа (ребро(дуга) и вершина) (В неориентированном графе если вершина инцидентна ребру то соответствующий элемент равен 1, в противном случае элемент равен 0).

Взвешенный граф — граф, в котором у каждого ребра и/или каждой вершины есть “вес” - некоторое число, которое может обозначать длину пути, его стоимость и т. п.



3 ТЕСТОВЫЕ ПРИМЕРЫ

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

3.1 Тест 1

Вход:

Необходимо дерево кратчайших путей в графе.

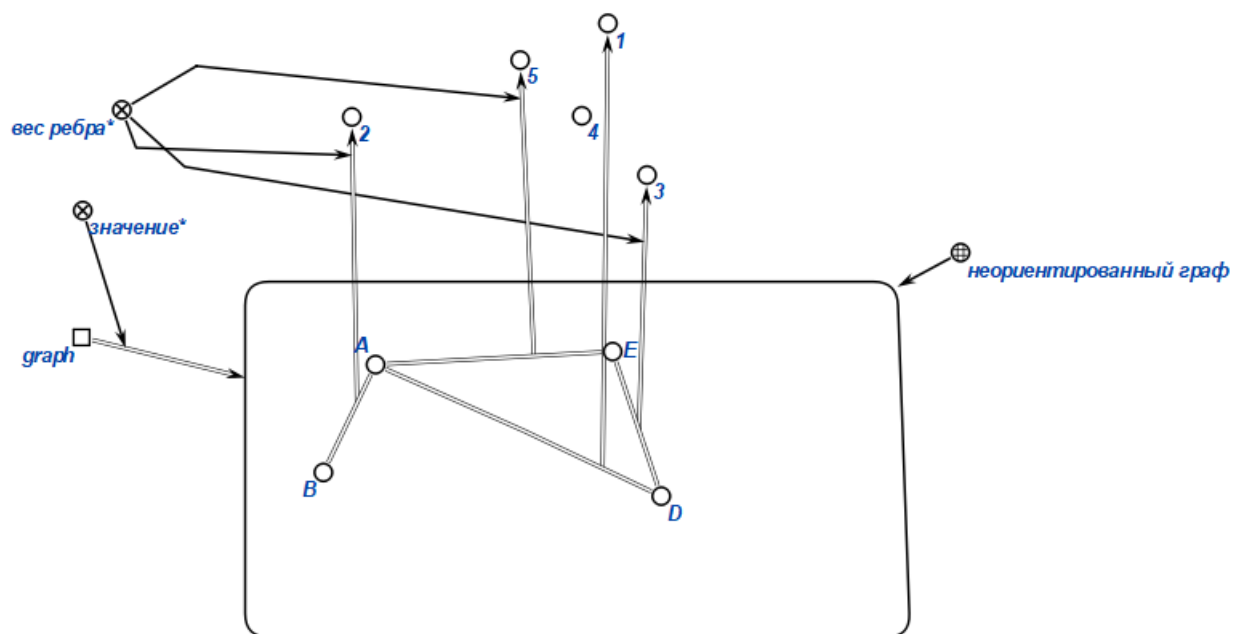


Рисунок 3.1 – Вход теста 1.1

Выход:

Будет выведен граф который является деревом кратчайших путей для исходного

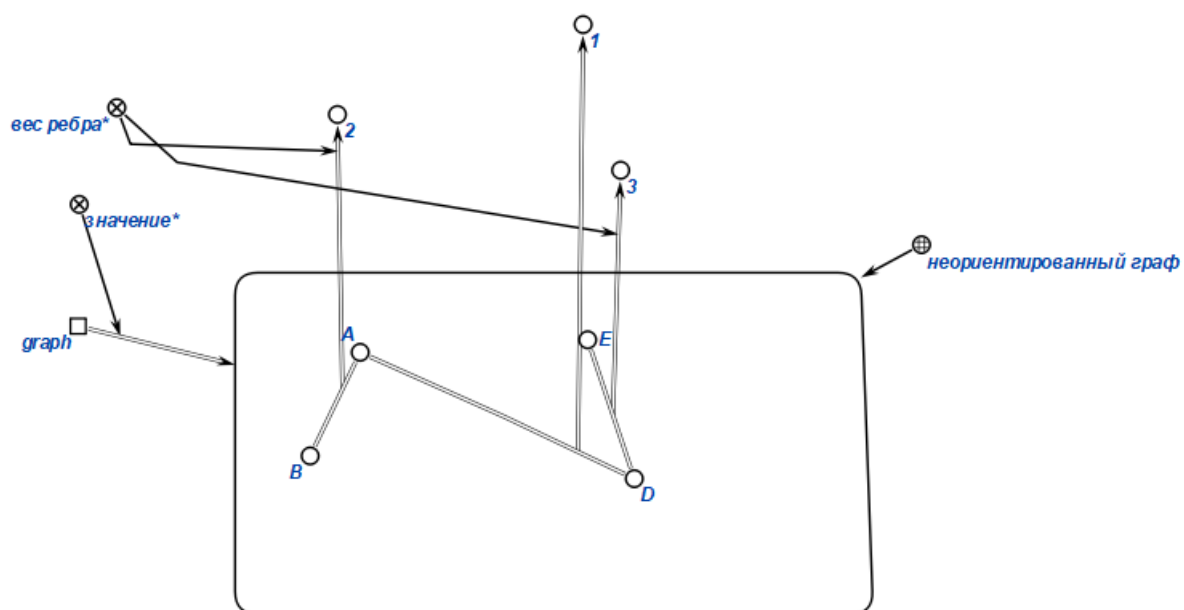


Рисунок 3.2 – Выход теста 1.2

3.2 Тест 2

Вход:

Необходимо дерево кратчайших путей в графе.

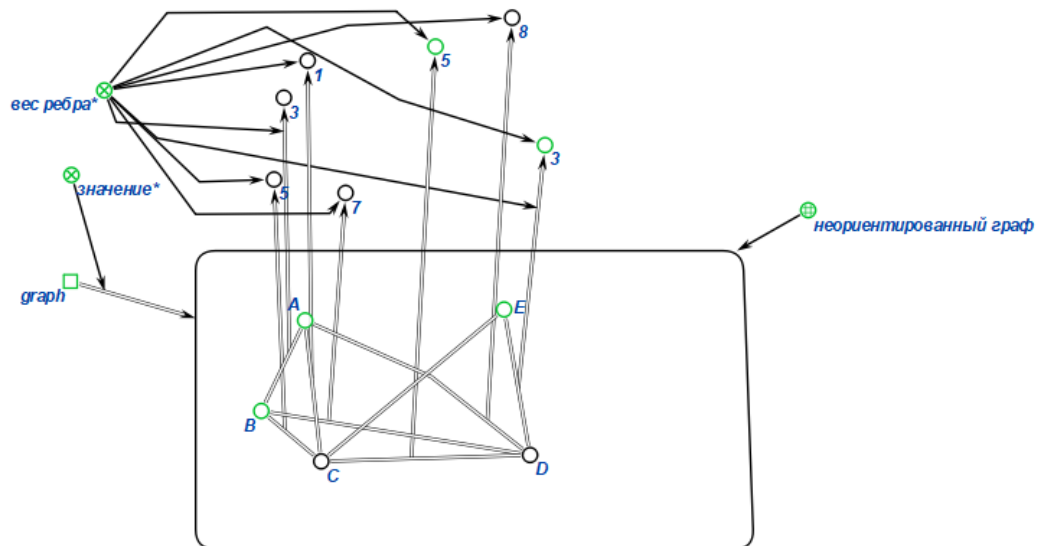


Рисунок 3.3 – Вход теста 2.1

Выход:

Будет выведен граф который является деревом кратчайших путей для исходного

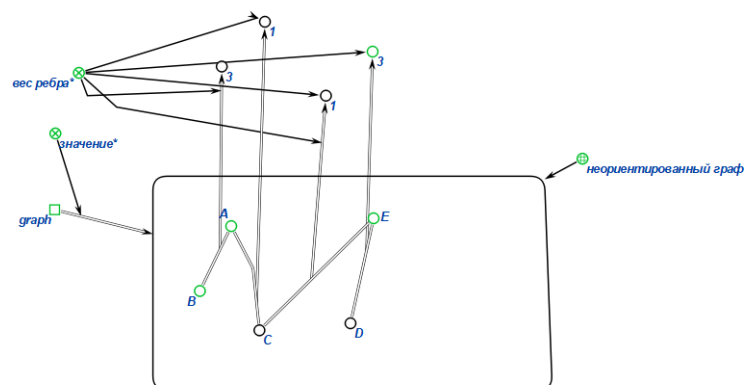


Рисунок 3.4 – Выход теста 2.2

3.3 Тест 3

Вход:

Необходимо дерево кратчайших путей в графе.

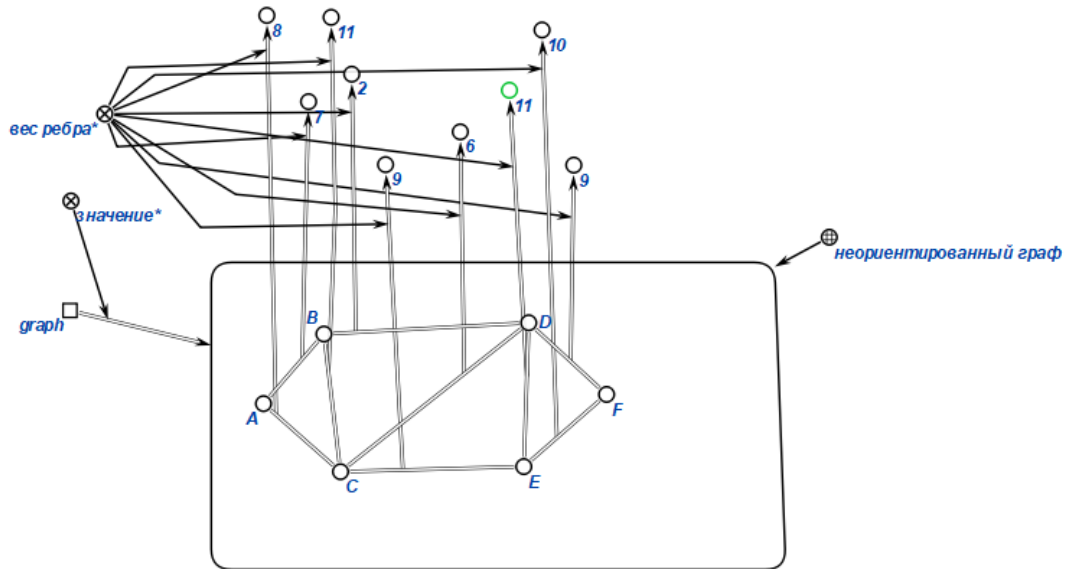


Рисунок 3.5 – Вход теста 3.1

Выход:

Будет выведен граф который является деревом кратчайших путей для исходного

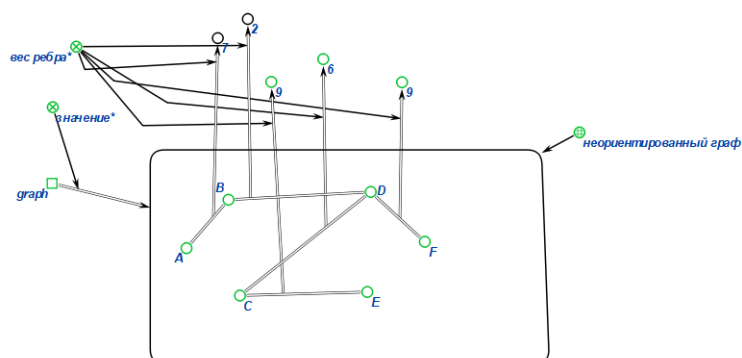


Рисунок 3.6 – Выход теста 3.2

3.4 Тест 4

Вход:

Необходимо дерево кратчайших путей в графе.

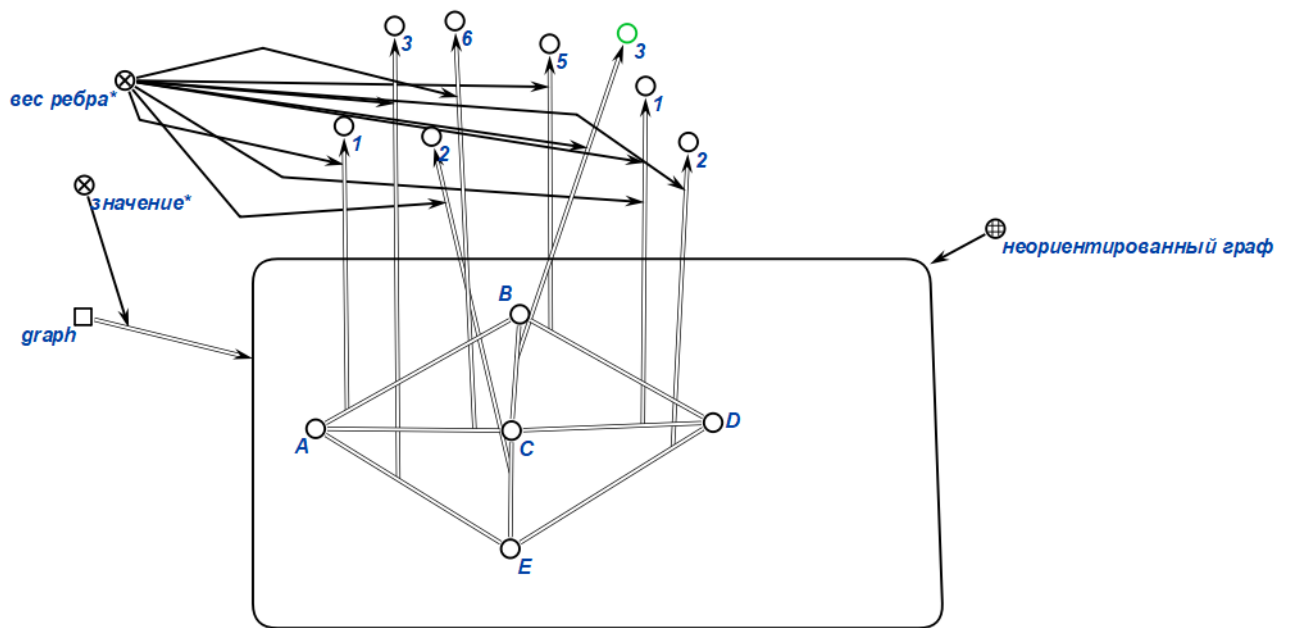


Рисунок 3.7 – Вход теста 4.1

Выход:

Будет выведен граф который является деревом кратчайших путей для исходного

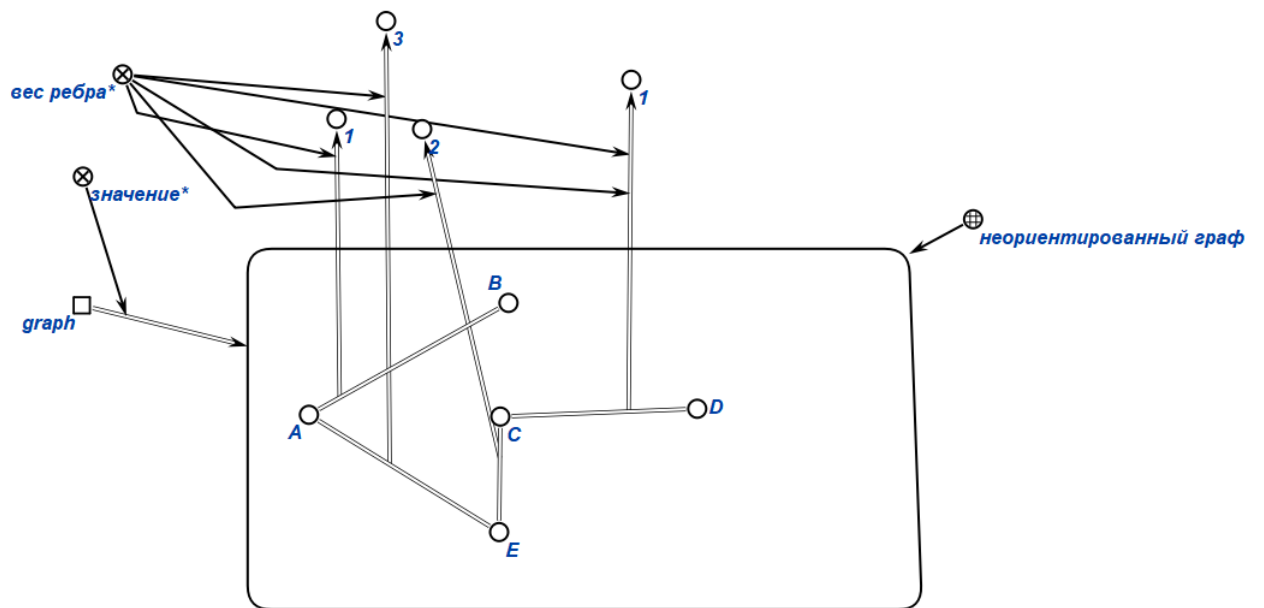


Рисунок 3.8 – Выход теста 4.2

3.5 Тест 5

Вход:

Необходимо дерево кратчайших путей в графе.

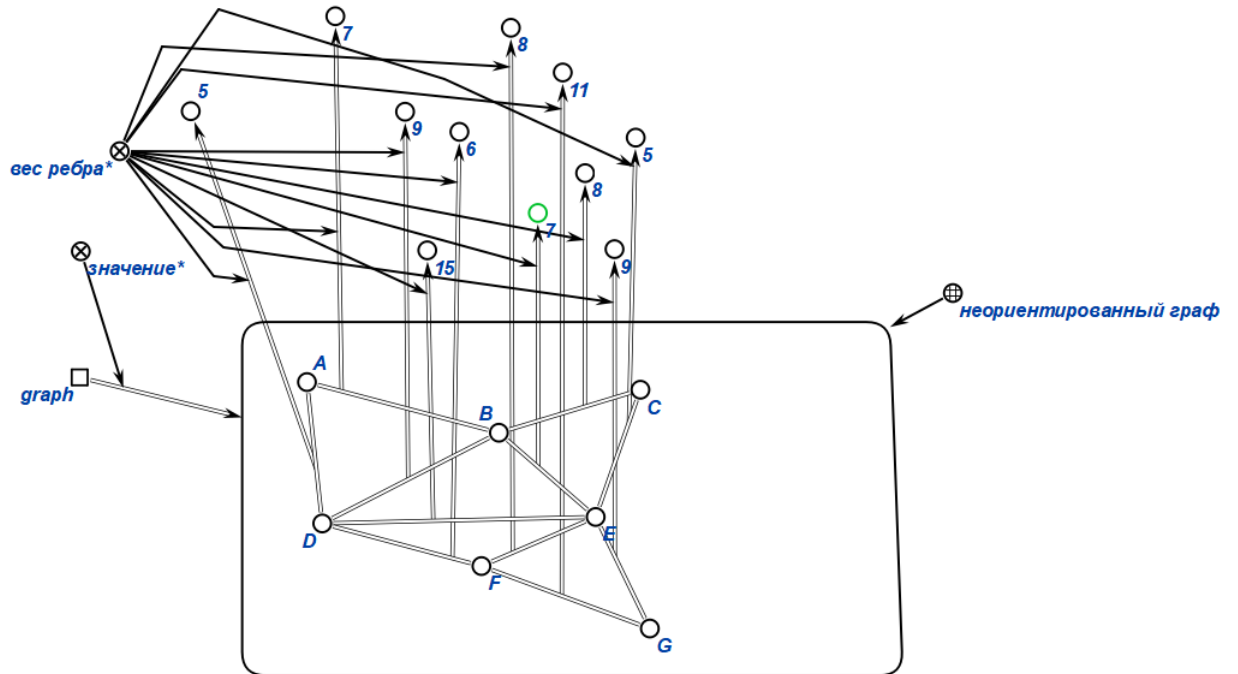


Рисунок 3.9 – Вход теста 5.1

Выход:

Будет выведен граф который является деревом кратчайших путей для исходного

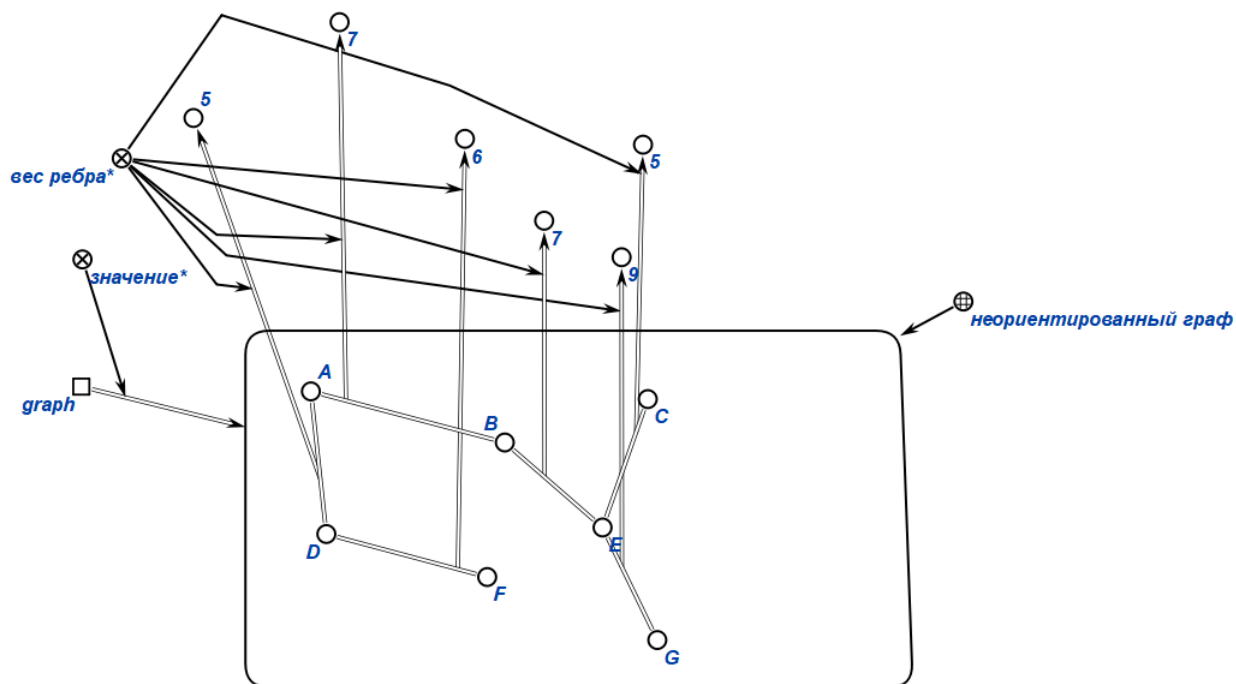


Рисунок 3.10 – Выход теста 5.2

4 ПРИМЕР РАБОТЫ АЛГОРИТМА В СЕМАНТИЧЕСКОЙ ПАМЯТИ

1. Шаг 1

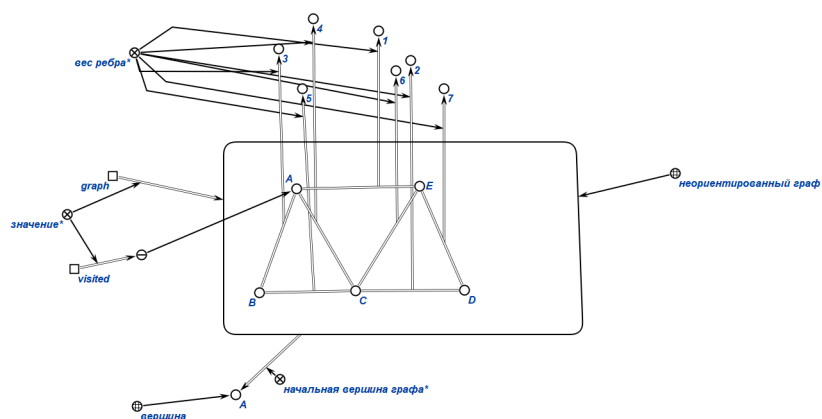


Рисунок 4.1 – Шаг 1

Задаем исходный взвешенный граф и начальную вершину. Создаем переменную `visited`, которая указывает на посещенную вершину графа.

2. Шаг 2

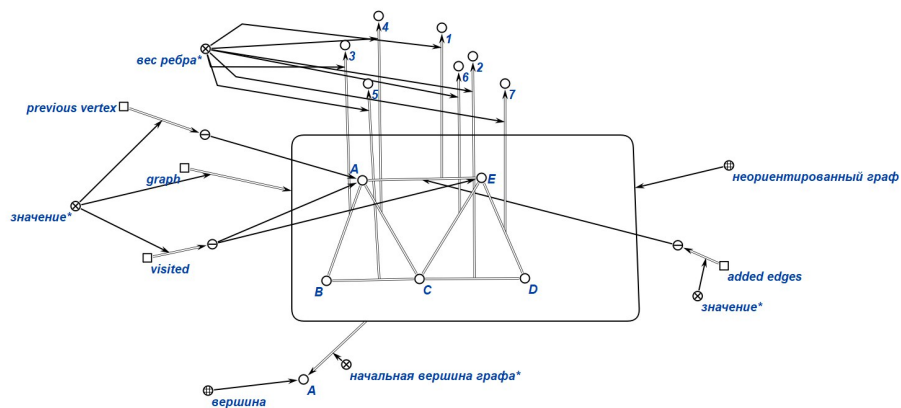


Рисунок 4.2 – Шаг 2

Добавляем ближайшую вершину в переменную visited. Добавляем ребро, смежное добавленной вершине, и вершине, ближайшей к ней (из уже добавленных) в переменную added edges. Создаём переменную previous vertex которой присваиваем предыдущую добавленную вершину.

3. Шаг 3

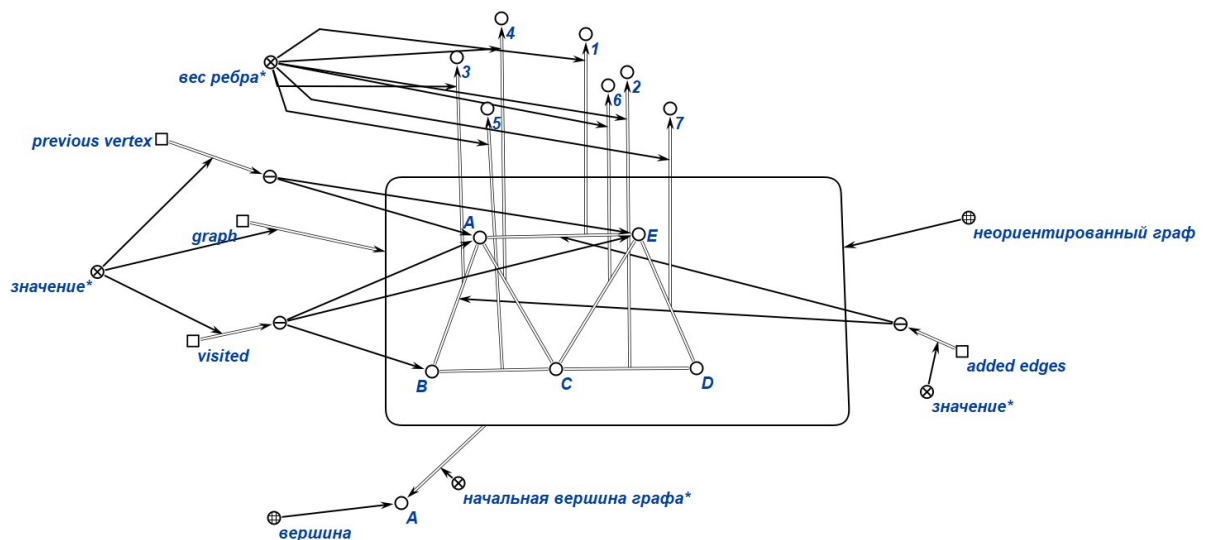
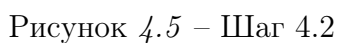
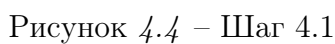


Рисунок 4.3 – Шаг 3

Смотрим, не является ли текущая ближайшая вершина смежной вершине в переменной previous vertex, в таком случае добавляем ее в переменную visited, добавляем ребро, смежное добавленной вершине и вершине в переменной previous vertex в переменную added edges. В противном случае добавляем ближайшую вершину в переменную visited. Добавляем ребро, смежное добавленной вершине, и вершине, ближайшей к ней (из уже добавленных) в

4. Шаг 4 Повторяем, пока не будут добавлены все вершины



10

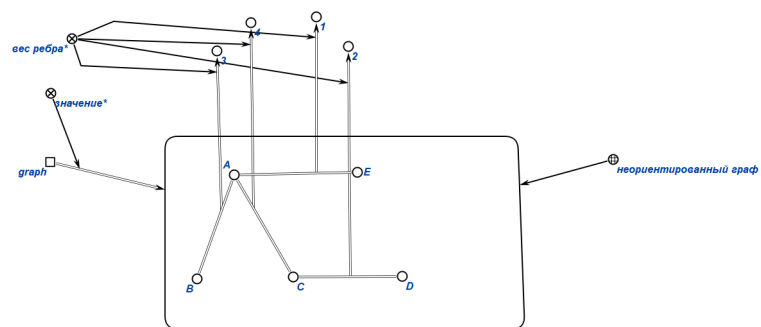


Рисунок 4.6 – Шаг 5

5 ЗАКЛЮЧЕНИЕ

Был построен фрагмент онтологии, продемонстрирована работа программы решения теоретико-графовой задачи по нахождению дерева кратчайших путей в неориентированном графе в семантической памяти.

6 ЛИТЕРАТУРА

- [1] Емеличев В. А., Мельников О. И., Сарванов В. И., Тышкевич Р. И. Лекции по теории графов. М.: Наука, 1990. 384с. (Изд.2, испр. М.: УРСС, 2009. 392 с.)
- [2] Оре О. Теория графов. – 2-е изд.. – М.: Наука, 1980. – С. 336.
- [3] Лазуркин Д.А., Руководство к выполнению расчетной работы по курсам ОИИ и ППВИС, 2013. 384с. (Минск)