

ПИиКТ

Системы искусственного интеллекта

Лабораторная работа №2

Вариант №2

Выполнил: Балтабаев Дамир

Группа: P33121

Преподаватель: Авдюшина А.Е.

**Задание**

Имеется транспортная сеть, связывающая города СНГ. Сеть представлена в виде таблицы связей между городами. Связи являются двусторонними, т.е. допускают движение в обоих направлениях. Необходимо проложить маршрут из одной заданной точки в другую.

Этап 1. Неинформированный поиск. На этом этапе известна только топология связей между городами. Выполнить:

1) поиск в ширину;

2) поиск глубину;

3) поиск с ограничением глубины;

4) поиск с итеративным углублением;

5) двунаправленный поиск.

Отобразить движение по дереву на его графе с указанием сложности каждого вида поиска. Сделать выводы.

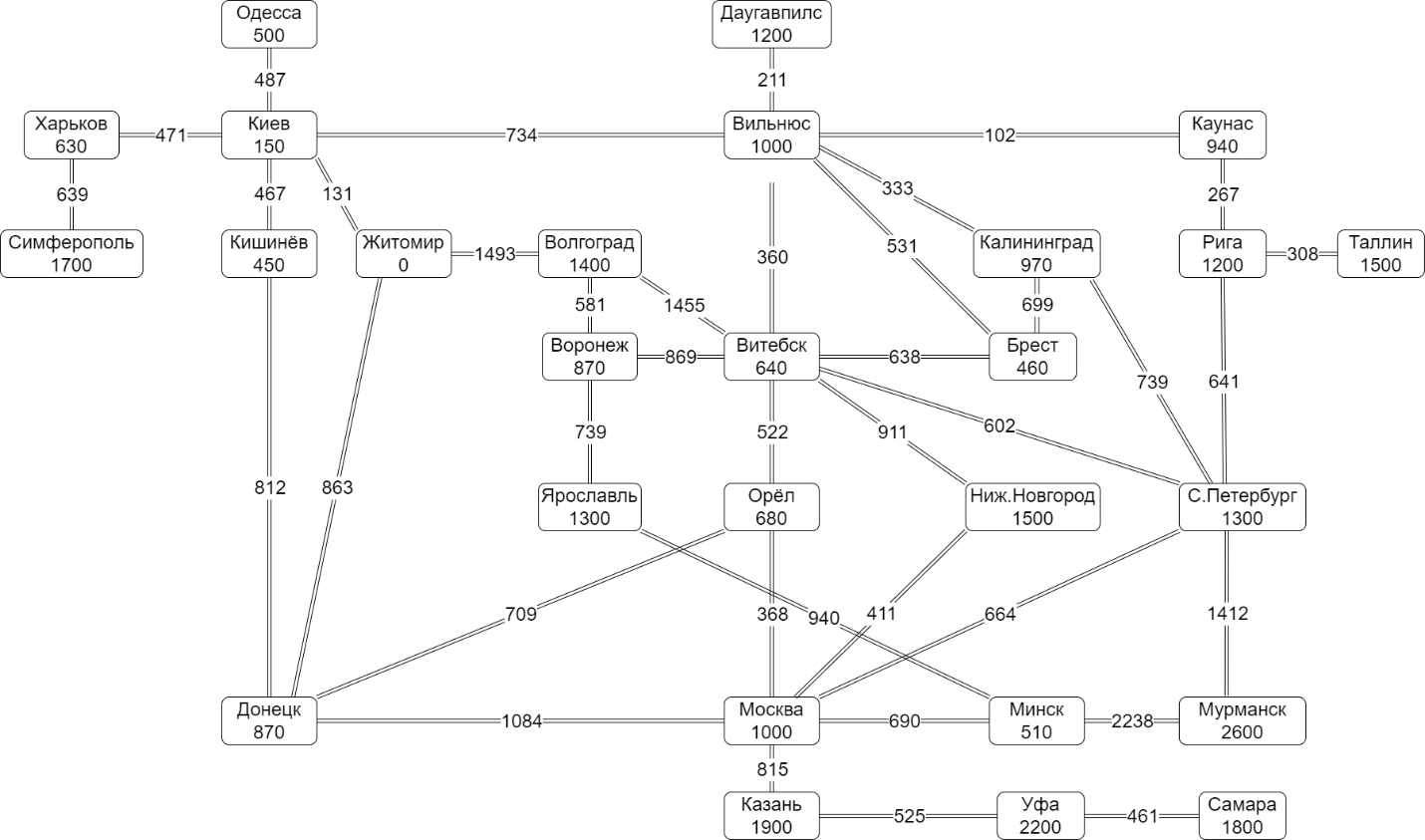
Этап 2. Информированный поиск. Воспользовавшись информацией о протяженности связей от текущего узла, выполнить:

1) жадный поиск по первому наилучшему соответствию;

2) затем, использую информацию о расстоянии до цели по прямой от каждого узла, выполнить поиск методом минимизации суммарной оценки А\*.

Отобразить на графе выбранный маршрут и сравнить его сложность с неинформированным поиском. Сделать выводы.

**Граф**

****

**Выполнение**

**Неинформированный поиск:**

Поиск в глубину (DFS):

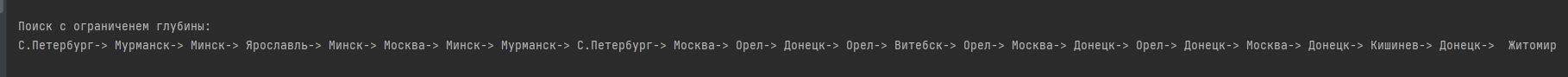


Поиск в ширину (BFS):

Изображение выглядит как текст

Автоматически созданное описание

Поиск с ограничением глубины при лимите = 3:



Поиск с итеративным углублением (1-5):

**Результат при лимите = 1**: Невозможно найти заданный путь: С.Петербург-> Мурманск-> С.Петербург-> Москва-> С.Петербург-> Рига-> С.Петербург-> Калининград-> С.Петербург-> Витебск-> С.Петербург->

**Результат при лимите = 2:** Невозможно найти заданный путь: С.Петербург-> Мурманск-> Минск-> Мурманск-> С.Петербург-> Москва-> Орел-> Москва-> Донецк-> Москва-> Минск-> Москва-> Ниж.Новгород-> Москва-> Казань-> Москва-> С.Петербург-> Москва-> С.Петербург-> Рига-> Таллинн-> Рига-> С.Петербург-> Рига-> Каунас-> Рига-> С.Петербург-> Калининград-> С.Петербург-> Калининград-> Брест-> Калининград-> Вильнюс-> Калининград-> С.Петербург-> Витебск-> Орел-> Витебск-> С.Петербург-> Витебск-> Ниж.Новгород-> Витебск-> Волгоград-> Витебск-> Воронеж-> Витебск-> Брест-> Витебск-> Вильнюс-> Витебск-> С.Петербург

**Результат при лимите = 3**: С.Петербург-> Мурманск-> Минск-> Ярославль-> Минск-> Москва-> Минск-> Мурманск-> С.Петербург-> Москва-> Орел-> Донецк-> Орел-> Витебск-> Орел-> Москва-> Донецк-> Орел-> Донецк-> Москва-> Донецк-> Кишинев-> Донецк-> Житомир

**Результат при лимите = 4:** С.Петербург-> Мурманск-> Минск-> Ярославль-> Воронеж-> Ярославль-> Минск-> Москва-> Орел-> Москва-> Донецк-> Москва-> Минск-> Москва-> Ниж.Новгород-> Москва-> Казань-> Москва-> Минск-> Мурманск-> С.Петербург-> Москва-> Орел-> Донецк-> Кишинев-> Донецк-> Житомир

**Результат при лимите = 5:** С.Петербург-> Мурманск-> Минск-> Ярославль-> Воронеж-> Волгоград-> Воронеж-> Витебск-> Воронеж-> Ярославль-> Минск-> Москва-> Орел-> Донецк-> Орел-> Витебск-> Орел-> Москва-> Донецк-> Орел-> Донецк-> Москва-> Донецк-> Кишинев-> Донецк-> Житомир

Двунаправленный поиск**:**

**Изображение выглядит как текст

Автоматически созданное описание**

Поиск в глубину не является оптимальным поиском, т.к из-за своей особенности часто бродит по графу.

Поиск в ширину является оптимальным, но потребляет много памяти

Поиск с ограничением глубины не всегда является оптимальным, он помогает избежать блужданий по графу, но не всегда можно найти ответ при заданном лимите

Поиск с итеративным углублением решает проблемы поиска с ограничением глубины

Двунаправленный поиск выполняет поиск сразу из двух точек (из начальной и из конечной), тем самым помогая сократить процесс поиска вдвое

b – коэффициент ветвления = 3

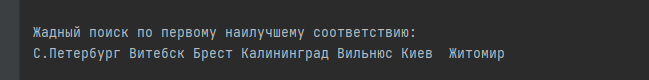
d – глубина самого поверхностного решения = 4

m – максимальная глубина дерева = 14

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Метод | Полнота | Временная сложность | Затраты памяти | Оптимальность |
| Поиск в ширину | Да | 3^5 | 3^5 | Да |
| Поиск в глубину | Нет | 3^14 | 3\*14 | Нет |
| Поиск с ограничением глубины | Нет | 3^e | 3\*e | Нет |
| Поиск с итеративным углублением | Да | 3^4 | 3\*4 | Да |
| Двунаправленный поиск | Да | 3^2 | 3^2 | Да |

**Информированный поиск:**

Жадный поиск по первому наилучшему соответствию:



Алгоритм A\*:

Изображение выглядит как текст

Автоматически созданное описание

Жадный поиск проигрывает поиску A\*, т.к не учитывает общий путь (протяженность), в отличие от A\*

**Вывод:**

В данной лабораторной работе я познакомился с информированными и неинформированными поисками, разобрался в их реализации и реализовал их программно.