

УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Системы искусственного интеллекта»

### **Лабораторная работа №3**

*Вариант 9*

Студент

*Патутин В.М*

*P33101*

Преподаватель

*Бессмертный И. А.*

Санкт-Петербург, 2021 г.

## Задание:

Исследование алгоритмов решения задач методом поиска.

Описание предметной области. 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:

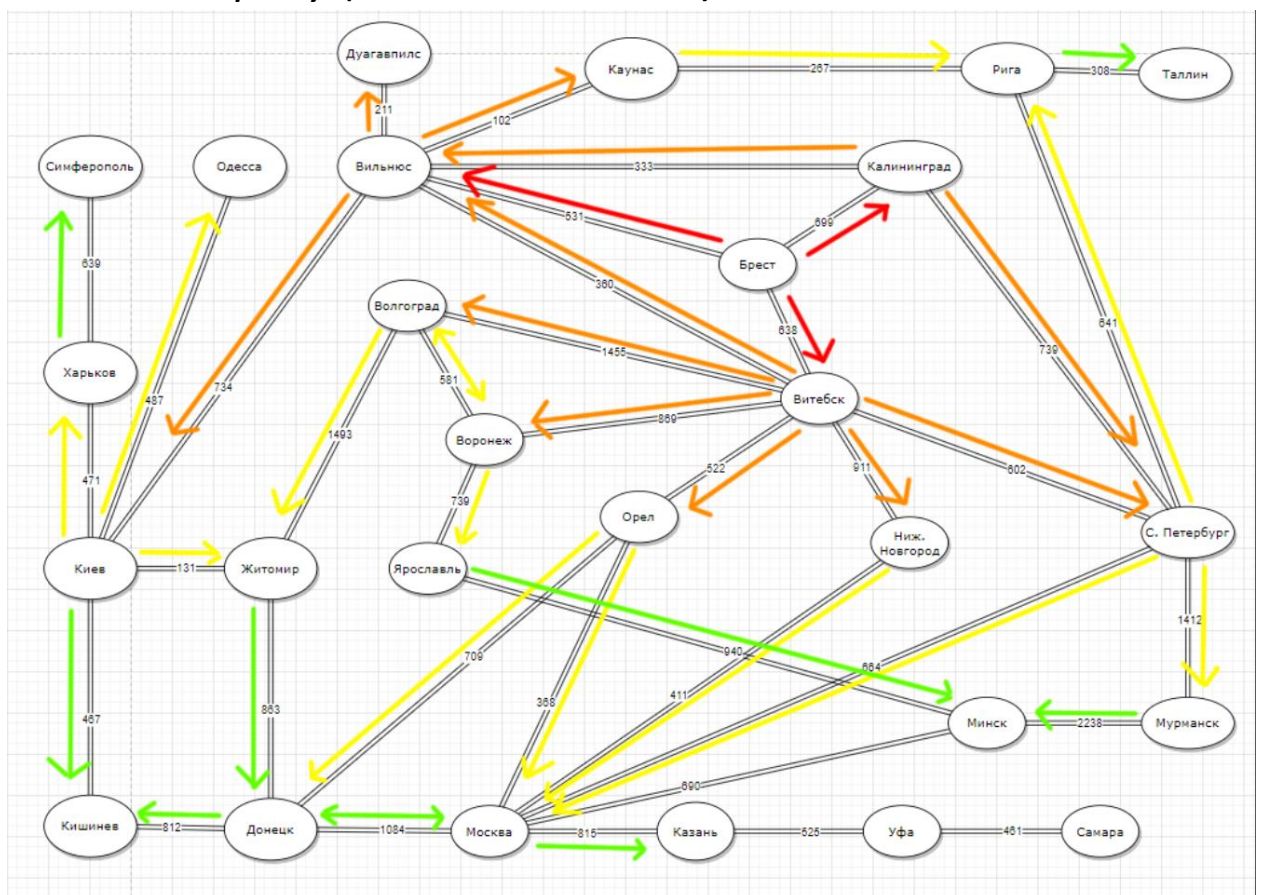
- 1) поиск в ширину;
- 2) поиск глубину;
- 3) поиск с ограничением глубины;
- 4) поиск с итеративным углублением;
- 5) двунаправленный поиск.

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

- 1) жадный поиск по первому наилучшему соответствию;
- 2) затем, используя информацию о расстоянии до цели по прямой от каждого узла, выполнить поиск методом минимизации суммарной оценки  $A^*$ .

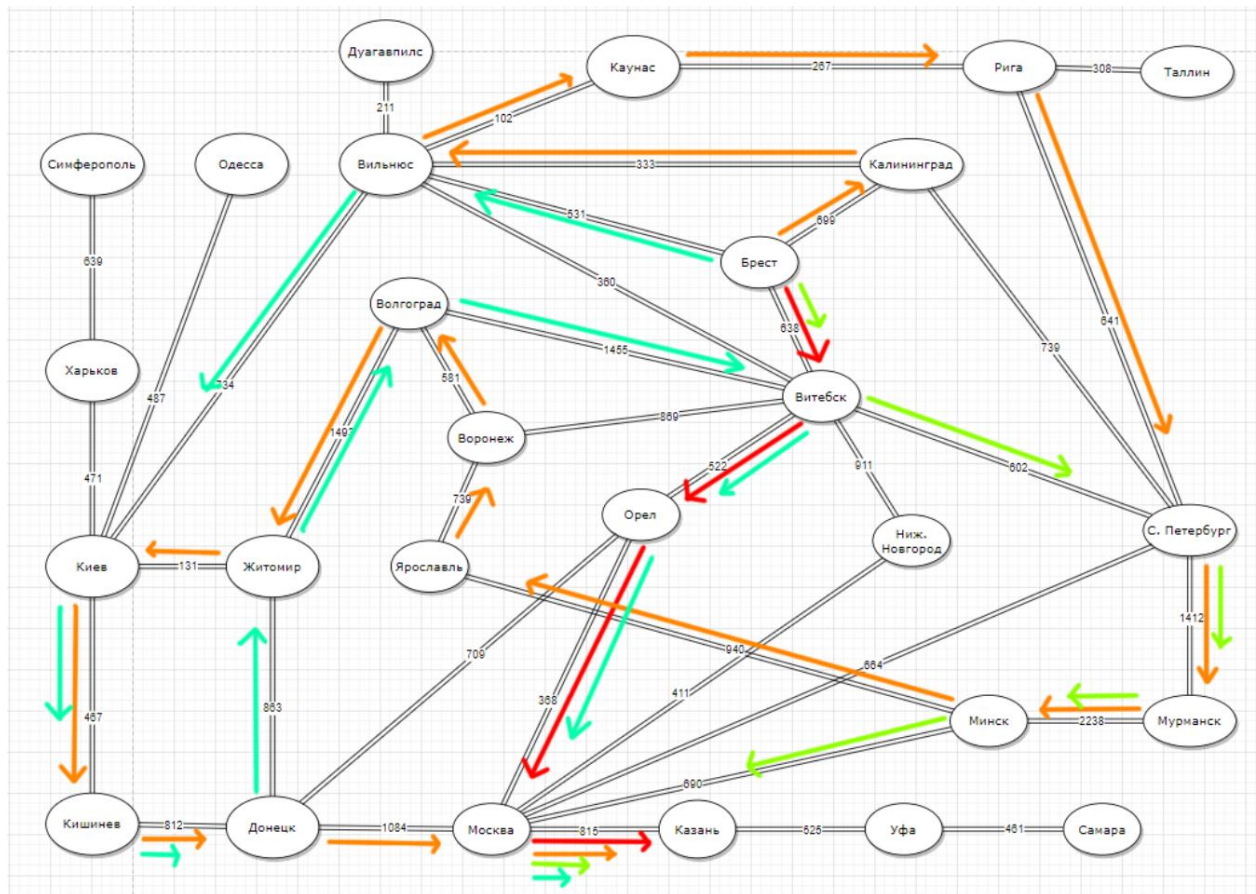
The diagram illustrates a network of cities and their connections. The nodes (cities) are: Дуагалис, Каунас, Рига, Таллин, Калининград, Брест, Витебск, С. Петербург, Ниж. Новгород, Минск, Мурманск, Уфа, Самара, Казань, Москва, Ярославль, Воронеж, Волгоград, Житомир, Киев, Харьков, Симферополь, Одесса, и Донецк. The edges (connections) are labeled with numerical values representing weights or distances. The network shows a high degree of connectivity, particularly around the central nodes like Витебск and С. Петербург.

Поиск в ширину (сложность  $b^{d+1}$ )

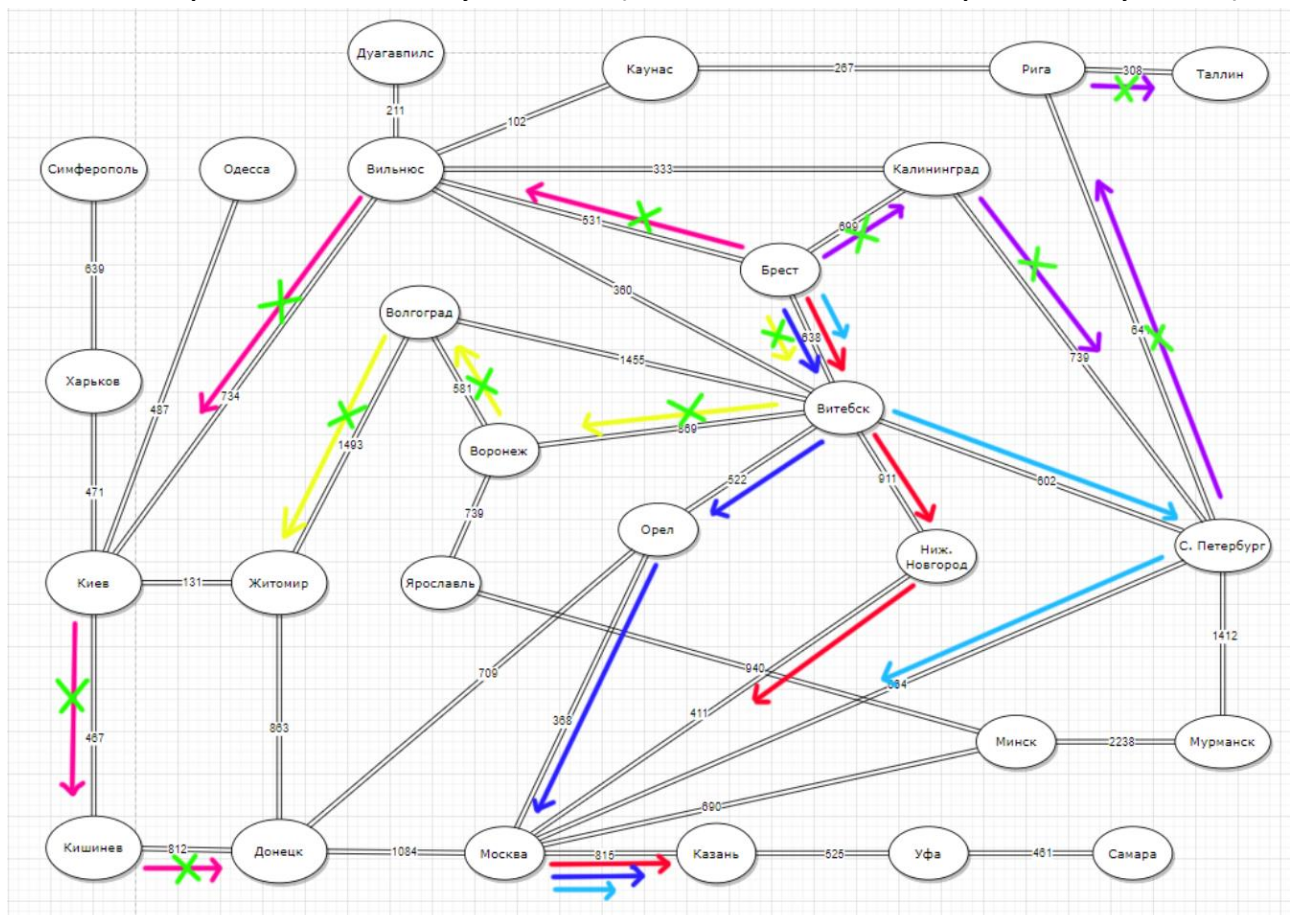




Поиск в глубину (сложность  $b^m$ )



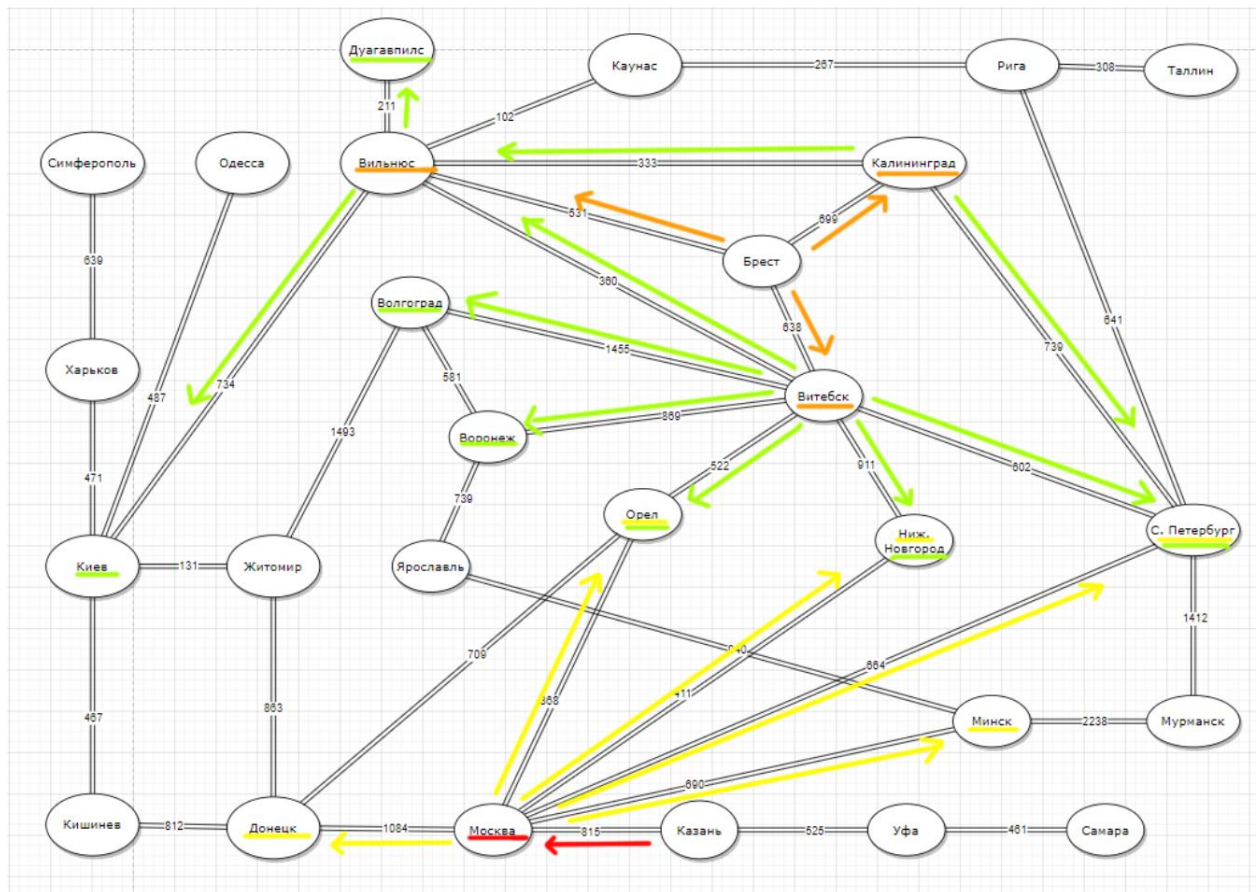
Поиск с ограничением глубины 4 (сложность  $b^e$ , e-предел глубины)







## Двунаправленный поиск (сложность $b^{d/2}$ )



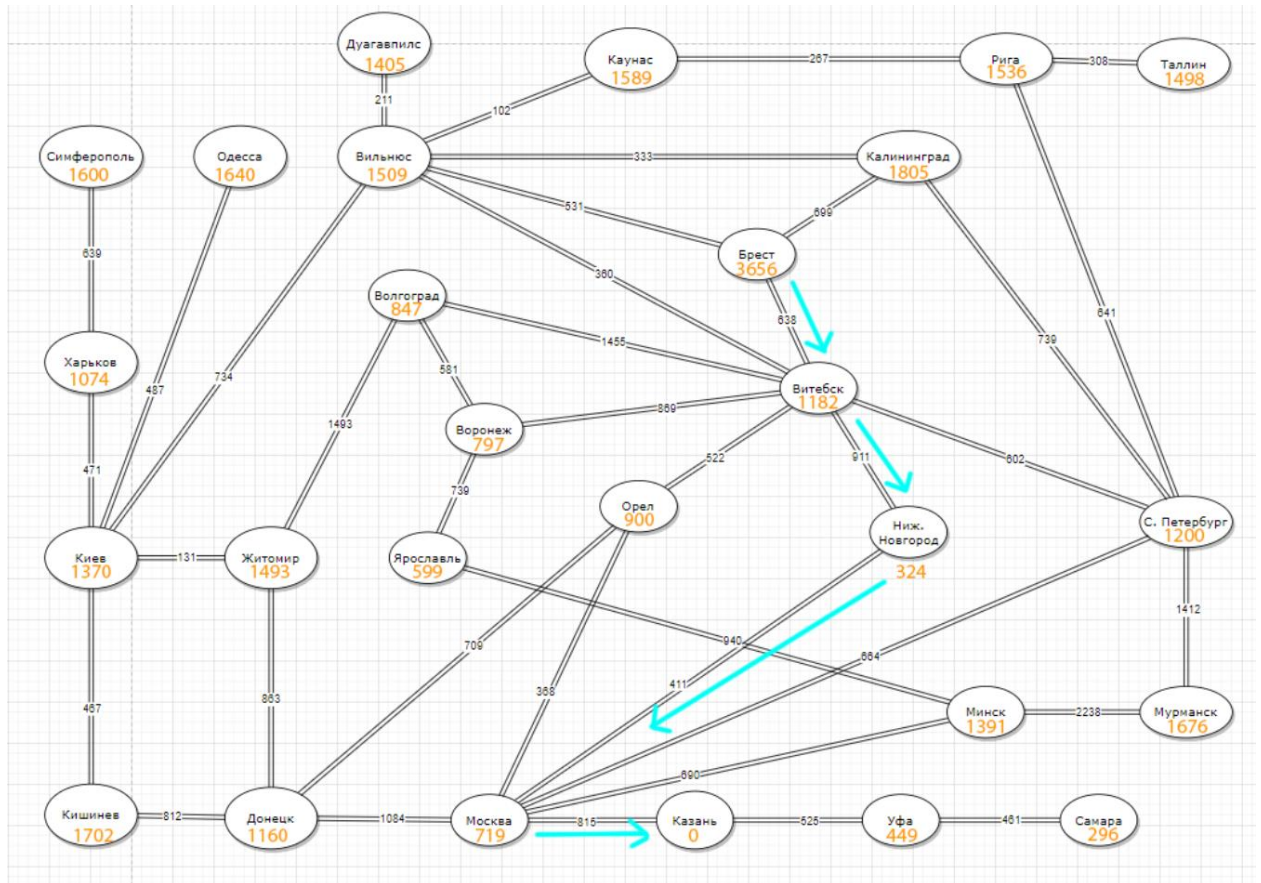
### Вывод Этапа 1:

Метод	Полнота	Временная сложность	Затраты памяти	Оптимальность
Поиск в ширину	Да	$b^{d+1}$	$b^{d+1}$	Да
Поиск по критерию стоимости	Да	$b^{1+C/n}$	$b^{1+C/n}$	Да
Поиск в глубину	Нет	$b^m$	$bm$	Нет
Поиск с ограничением глубины	Нет	$b^e$	$be$	Нет
Поиск с итеративным углублением	Да	$b^d$	$bd$	Да
Двунаправленный поиск	Да	$b^{d/2}$	$b^{d/2}$	Да

Алгоритм поиска в ширину не является оптимальным, так как имеет большую сложность, из-за запоминания всех вершин, но при этом всегда находит цель. Поиск в глубину может дать неверный вывод, зайдя в тупик. Также он может долго углубляться в один вариант решения, не проверив короткий. Поиск с ограничением глубины вводит дополнительное условие неполноты, что ускоряет процесс, но имеет шанс не найти цель. Двунаправленный поиск является оптимальным.

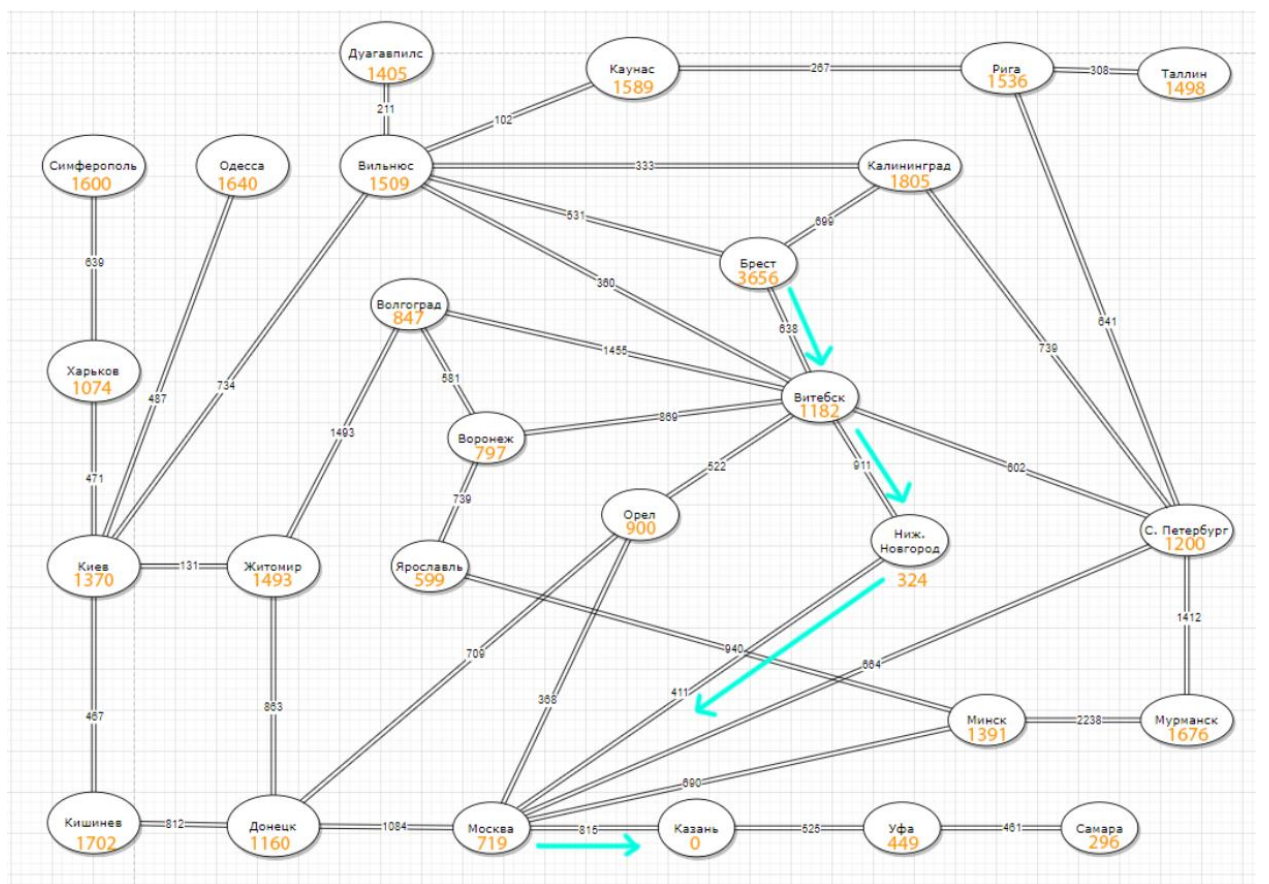
Этап 2:

Жадный поиск по первому наилучшему  
соответствию (сложность  $b^m$ )



```

graph TD
    Брест([Брест]) --> Вильнюс1([Вильнюс])
    Брест --> Калининград([Калининград])
    Брест --> Витебск([Витебск])
    Вильнюс1 --- V1["531+(1509) = 2040"]
    Калининград --- K["699+(1805) = 2504"]
    Витебск --- В1["638+(1182) = 1820"]
    Витебск --> Вильнюс2([Вильнюс])
    Витебск --> Волгоград([Волгоград])
    Витебск --> Воронеж([Воронеж])
    Витебск --> Орел([Орел])
    Витебск --> НижНовгород([Ниж. Новгород])
    Витебск --> СанктПетербург([Санкт-Петербург])
    Вильнюс2 --- V2["638+360+(1509) = 2507"]
    Волгоград --- V3["638+1455+(847) = 2940"]
    Воронеж --- V4["638+869+(797) = 2304"]
    Орел --- V5["638+522+(900) = 2060"]
    НижНовгород --- V6["638+911+(324) = 1873"]
    СанктПетербург --- V7["638+602+(1200) = 2440"]
    НижНовгород --> Москва([Москва])
    Москва --- M["638+911+411+(719) = 2679"]
    Москва --> Донецк([Донецк])
    Москва --> Казань([Казань])
    Донецк --- D["638+911+411+1084+(1160) = 4204"]
    Казань --- K2["638+911+411+815 = 2775"]
  
```





## Вывод Этапа 2:

Жадный поиск по первому наилучшему соответствию	Нет	$b^*m$	$b^*m$	Нет
Поиск методом минимизации суммарной оценки $A^*$	Да	$b^*a$	$a^*m$	Да

Алгоритм жадного поиска не является оптимальным, из-за возможного нахождения город близко, но при этом добраться напрямую нельзя.

Метод  $A^*$  является оптимальным, при условии, что выбрана допустимая эвристическая функция, которая не переоценивает стоимость. Но поиск  $A^*$  не является применимым на практике из-за того, что ресурсы пространства исчерпываются намного быстрее чем временные.

## Выводы

При выполнении лабораторной работы я изучил работу алгоритмов поиска на информированном и неинформированном графе, а также сравнил их работу и сложность.