

Университет ИТМО  
Факультет программной инженерии и компьютерной техники

**Лабораторная работа №2**  
по «Системам искусственного интеллекта»  
Вариант 8

Выполнил:  
Студент группы Р33121  
Нуруллаев Даниил

Преподаватели:  
Авдюшина А.Е.  
Бессмертный И.А.

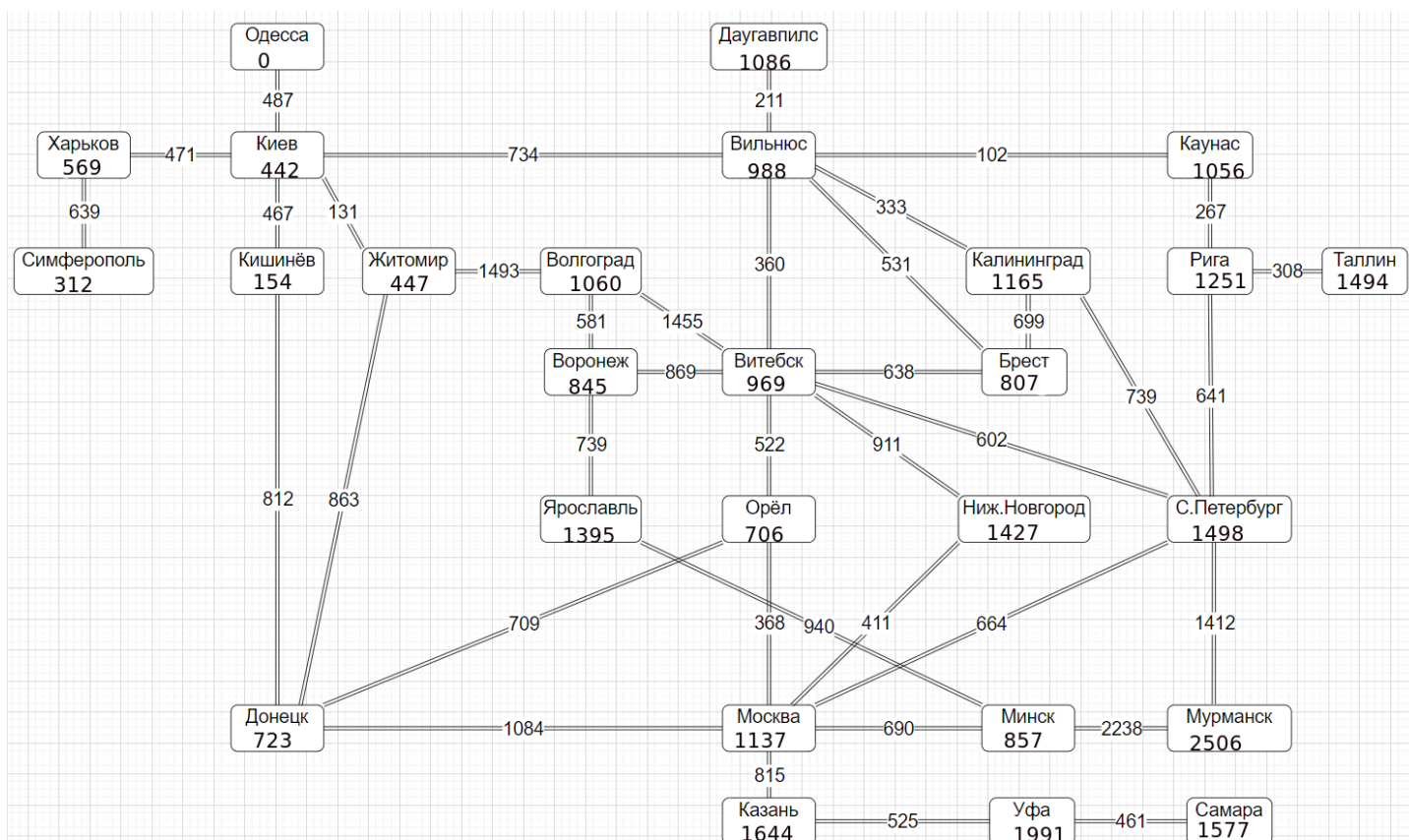
Санкт-Петербург  
2022

## 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

## Выполнение

Исходный граф:



## Неинформированный поиск:

## 1)Поиск в ширину

```

Вильнюс:[Брест, Витебск, Даугавпилс, Калининград, Каунас, Киев]
Брест:[Вильнюс, Витебск, Калининград]
Витебск:[Брест, Вильнюс, Воронеж, Волгоград, Ниж.Новгород, С.Петербург, Орел]
Даугавпилс:[Вильнюс]
Калининград:[Брест, Вильнюс, С.Петербург]
Каунас:[Вильнюс, Рига]
Киев:[Вильнюс, Житомир, Кишинев, Одесса, Харьков]

Путь :Вильнюс Киев Одесса

```

## 2) Поиск в глубину

```
Вильнюс->Брест->Витебск->Воронеж->Волгоград->Житомир->Киев->Кишинев->Донецк->Москва->Казань->Уфа->Самара->Возврат на Уфа->Возврат на Казань->Возврат на Москва->Ниж.Новгород->Возврат на Москва->Минск->Мурманск->С.Петербург->Калининград->Возврат на С.Петербург->Рига->Каунас->Возврат на Рига->Таллинн->Возврат на Рига->Возврат на С.Петербург->Возврат на Мурманск->Возврат на Минск->Ярославль->Возврат на Минск->Возврат на Москва->Орел->Возврат на Москва->Возврат на Донецк->Возврат на Кишинев->Возврат на Киев->Одесса
```

## 3) Поиск с ограничением глубины при $e=5$

```
Вильнюс deer0->Брест deer1->Витебск deer2->Воронеж deer3->Волгоград deer4->Житомир deer5->Возврат на Волгоград->Возврат на Воронеж->Ярославль deer4->Минск deer5->Возврат на Ярославль->Возврат на Воронеж->Возврат на Витебск->Ниж.Новгород deer3->Москва deer4->Казань deer5->Возврат на Москва->Донецк deer5->Возврат на Москва->С.Петербург deer5->Возврат на Москва->Орел deer5->Возврат на Москва->Возврат на Ниж.Новгород->Возврат на Витебск->Возврат на Брест->Калининград deer2->Возврат на Брест->Возврат на Вильнюс->Даугавпилс deer1->Возврат на Вильнюс->Каунас deer1->Рига deer2->Таллинн deer3->Возврат на Рига->Возврат на Каунас->Возврат на Вильнюс->Киев deer1->Кишинев deer2->Возврат на Киев->Одесса deer2
```

## 4) Поиск с итеративным углублением при $e=1$ to 4

```
Вильнюс deer0->Брест deer1->Возврат на Вильнюс->Витебск deer1->Возврат на Вильнюс->Даугавпилс deer1->Возврат на Вильнюс->Калининград deer1->Возврат на Вильнюс->Каунас deer1->Возврат на Вильнюс->Киев deer1->Возврат на Вильнюс->Нельзя добраться!  
Вильнюс deer0->Брест deer1->Витебск deer2->Возврат на Брест->Калининград deer2->Возврат на Брест->Возврат на Вильнюс->Даугавпилс deer1->Возврат на Вильнюс->Каунас deer1->Рига deer2->Возврат на Каунас->Возврат на Вильнюс->Киев deer1->Житомир deer2->Возврат на Киев->Кишинев deer2->Возврат на Киев->Одесса deer2  
Вильнюс deer0->Брест deer1->Витебск deer2->Воронеж deer3->Возврат на Витебск->Волгоград deer3->Возврат на Витебск->Ниж.Новгород deer3->Возврат на Витебск->С.Петербург deer3->Возврат на Витебск->Орел deer3->Возврат на Витебск->Возврат на Брест->Калининград deer2->Возврат на Брест->Возврат на Вильнюс->Даугавпилс deer1->Возврат на Вильнюс->Каунас deer1->Рига deer2->Таллинн deer3->Возврат на Рига->Возврат на Каунас->Возврат на Вильнюс->Киев deer1->Житомир deer2->Донецк deer3->Возврат на Житомир->Возврат на Киев->Кишинев deer2->Возврат на Киев->Одесса deer2  
Вильнюс deer0->Брест deer1->Витебск deer2->Воронеж deer3->Волгоград deer4->Возврат на Воронеж->Ярославль deer4->Возврат на Воронеж->Возврат на Витебск->Ниж.Новгород deer3->Москва deer4->Возврат на Ниж.Новгород->Возврат на Витебск->С.Петербург deer3->Калининград deer4->Возврат на С.Петербург->Рига deer4->Возврат на С.Петербург->Мурманск deer4->Возврат на С.Петербург->Возврат на Витебск->Орел deer3->Донецк deer4->Возврат на Орел->Возврат на Витебск->Возврат на Брест->Возврат на Вильнюс->Даугавпилс deer1->Возврат на Вильнюс->Каунас deer1->Возврат на Вильнюс->Киев deer1->Житомир deer2->Возврат на Киев->Кишинев deer2->Возврат на Киев->Одесса deer2
```

## 5) Двухнаправленный поиск

```
Вильнюс Киев Одесса
```

Поиск в ширину потребляет много памяти, но при этом является оптимальным способом обхода, в то время как поиск в глубину не всегда находит кратчайший путь в невзвешенном графе. Поиск с ограничением глубины позволяет задать глубину и присечь ненужные блуждания по графу, но при этом не всегда можем найти путь из-за нехватки глубины. Поиск с итеративным углублением решает

проблему предыдущего. Двухнаправленный поиск в ширину позволяет сократить количество время на выполнения в два раза.

Коэффициент ветвления = 3

Метод	Полнота	Временная сложность	Затраты памяти	Оптимальность
Поиск в ширину	Да	$3^3$	$3^3$	Да
Поиск в глубину	Нет	$3^{13}$	$3 \cdot 13$	Нет
Поиск с ограничением глубины	Нет	$3^e$	$3 \cdot e$	Нет
Поиск с итеративным углублением	Да	$3^2$	$3 \cdot 2$	Да
Двухнаправленный поиск	Да	$3^1$	$3^1$	Да

### Информированный поиск:

6) Жадный алгоритм

```
[Вильнюс, Киев, Одесса]
```

7) A\*

```
Вильнюс
Киев
Одесса
Количество км= 1221.0
```

Жадный алгоритм не оптимален так как не учитывает общую протяженность маршрута, в то время как A\* учитывает и весь маршрут и эвристическую функцию.

### Вывод

В данной лабораторной работе я познакомился с большим количеством алгоритмов поиска. Реализовал их в виде программного кода и разобрался в недостатках и преимуществах данных алгоритмов.