НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Проектный семинар
2 курс

Идрисов Дамир, Буев Антон, Раздьяконов Егор,
Голубев Никита

МИЭМ, Прикладная математика, БПМ-172

Планировщик маршрута для поездок на автомобиле

Содержание

1. Описание задачи
2. Начальные условия
3. Описание решения
4. Анализ результатов

Задача: найти оптимальный маршрут для посещения всех заданных пользователем пунктов и вывести его на карту.

Проектный семинар

Дано:

- Сетка из дорог(дорожных путей), представленная в виде взвешенного связного графа;
- Количество пунктов назначения, которые хочет посетить пользователь;
- Координаты пунктов;
- Начальный и конечный пункт.

Проблема кратчайшего пути

Дорожную сеть можно легко представить в виде графа, то есть в виде набора узлы V (перекрестки) и ребра E (отрезки дороги), где каждое ребро соединяет два узла. Каждому ребру присваивается вес, например длина дороги или оценка времени, необходимого для проезда по дороге.

Проектный семинар

В теории графов

Вычисление кратчайших путей между двумя узлами является классической задачей.

На самом деле, мы можем различить несколько вариантов этой проблемы:

• point-to-point: вычисление длины кратчайшего пути от заданного исходного узла
$s \in V$ до заданного целевого узла $t \in V$;

• single-source: для данного исходного узла $s \in V$ вычислим кратчайший путь длины до всех узлов $v \in V$;

• many-to-many: для заданных наборов узлов $S, T \subseteq V$ вычислить кратчайший путь
длина для каждой пары узлов $(s, t) \in S \times T$;

• all-pairs: особый случай варианта many-to-many с $S := T := V$.

Проектный семинар

Начиная с исходного узла s в качестве корневого, растет дерево кратчайшего пути который содержит кратчайшие пути от начального узла до всех других.

Во время этого процесса каждый узел графа не достигнут, достигнут, или решен.

Узел, который уже принадлежит дереву - решен Если узел u решен, то кратчайший путь от s до u известен.

Узел, который находится рядом с решенным узлом - достигнут. Если узел u достигнут, был найден путь от s до u, который может быть не самым коротким, и предварительное расстояние известно. Узел u, который не был посещен - не достигнут; для такого узла имеем расстояние = ∞.

Достигнутые узлы но не решены, управляются в очереди приоритетов. Приоритет узла u в очереди приоритетов находится предварительное расстояние. Достигнуты, но не решены узлы также называются узлами, находящимися в очереди.

Проектный семинар

Первоначально s вставляется в приоритетную очередь с ориентировочным расстоянием 0. Таким образом, s достигнут, все остальные узлы не достигнуты.

Пока очередь приоритетов не пуста, удаляется узел u с наименьшим предварительным расстоянием и добавляется в дерево кратчайшего пути, т. е. u решен. Кроме того, состояние крайних узлов проверяются:

• если ребро (u, v) приводит к не достигнутому узлу v, v добавляется к приоритету

очередь; теперь v достигнута;

• если ребро (u, v) приводит к узлу v, находящемуся в очереди, состояние v в очереди приоритетов обновляется при условии, что длина пути от s до u меньше старого значения v;

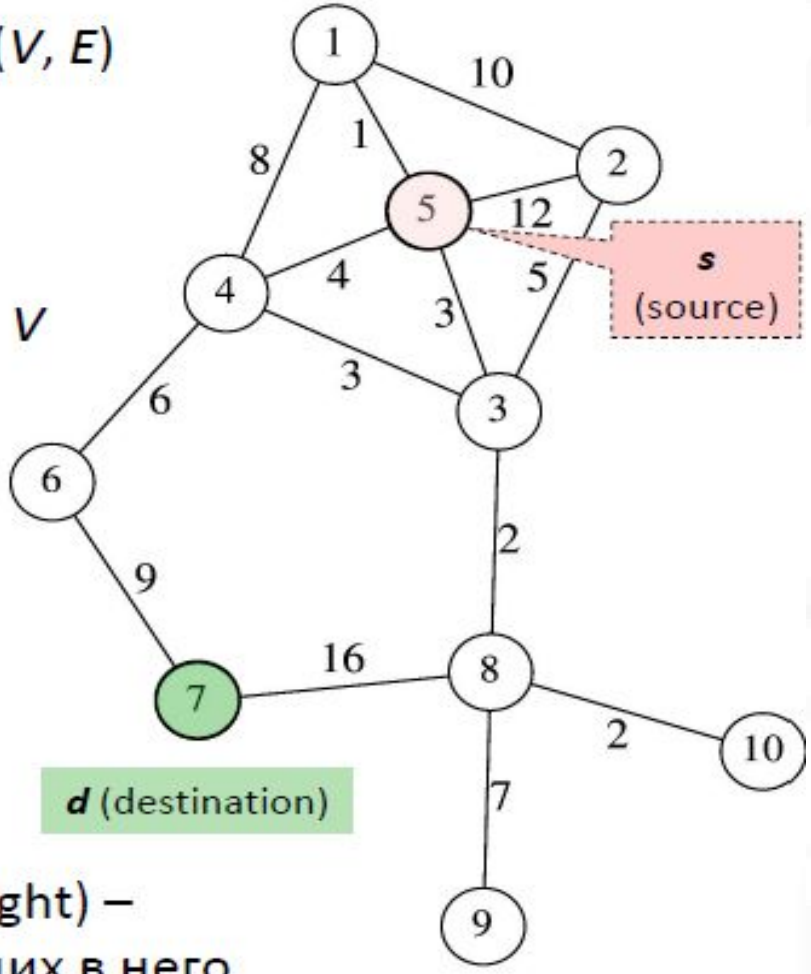
• если ребро (u, v) приводит к решенному узлу v, оно игнорируется.

Алгоритм Дейкстры

Алгоритм Дейкстры — алгоритм на графах, изобретённый нидерландским учёным Эдсгером Дейкстрой в 1959 году. Находит кратчайшие пути от одной из вершин графа до всех остальных.

Поиск кратчайшего пути в графе

- **Имеется** взвешенный граф $G = (V, E)$
- Каждому ребру $(i, j) \in E$ назначен вес w_{ij}
- Заданы начальная вершина $s \in V$ и конечная $d \in V$
- **Требуется** найти кратчайший путь из вершины s в вершину d (shortest path problem)
- Длина пути (path length, path cost, path weight) – это сумма весов ребер, входящих в него



Описание программы

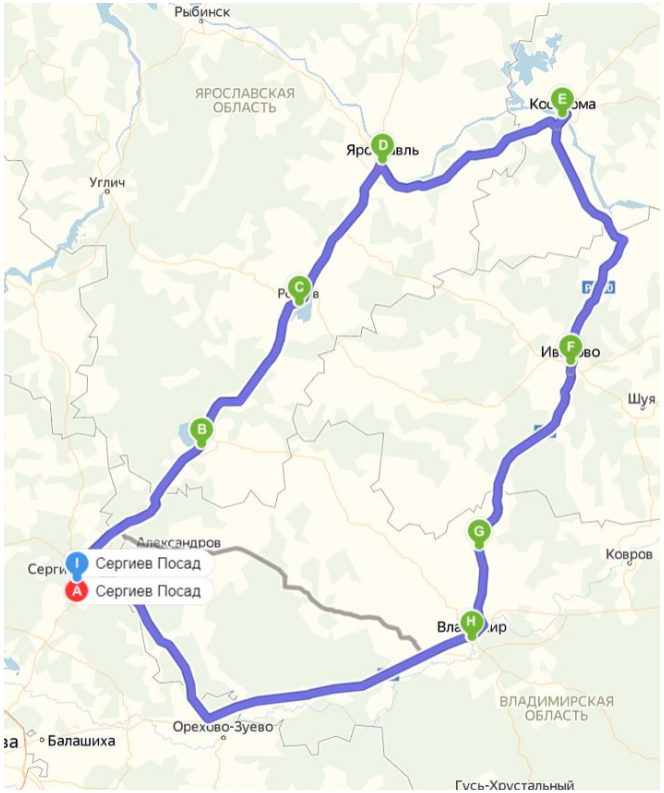
На вход подается массив из координат пунктов, которые пользователь хочет посетить. Программа составляет кратчайший маршрут из первого пункта в последний, меняя очередность посещения промежуточных пунктов и выбирая кратчайшую дорогу между соседними пунктами.

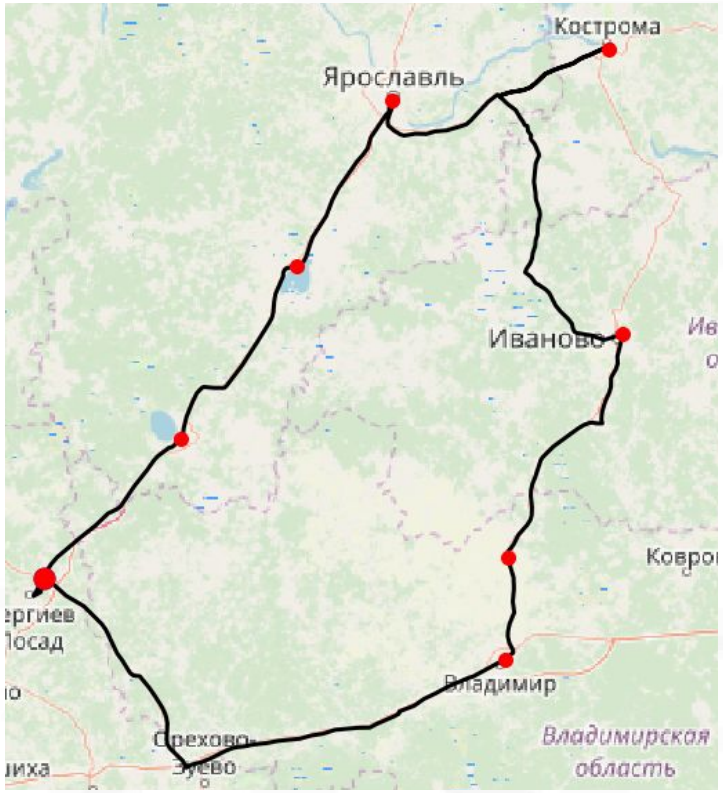
Проектный семинар

Программа была написана на Python с использованием Jupyter Notebook и следующих библиотек:

- NumPy
- NetworkX
- Json
- Matplotlib
- Smopy (модифицированный)

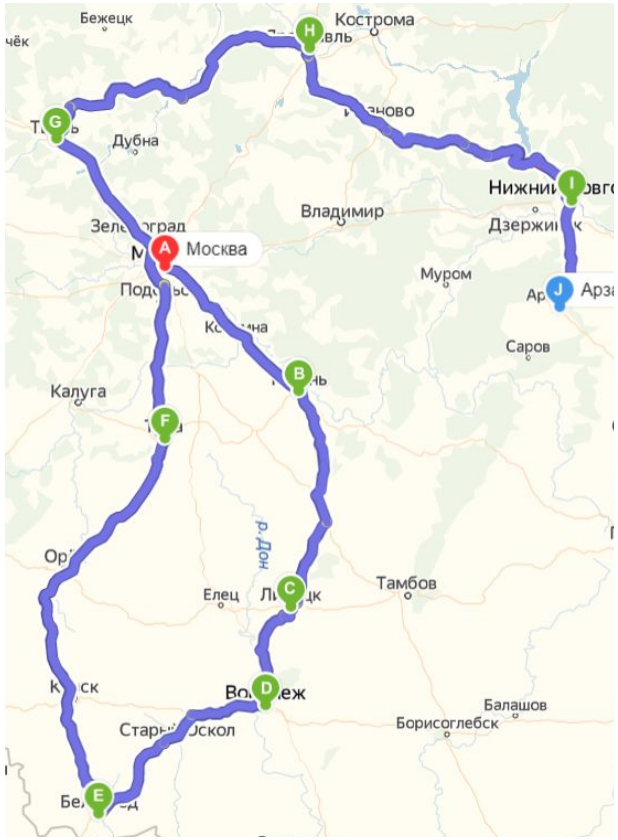
Маршрут “Золотое кольцо”





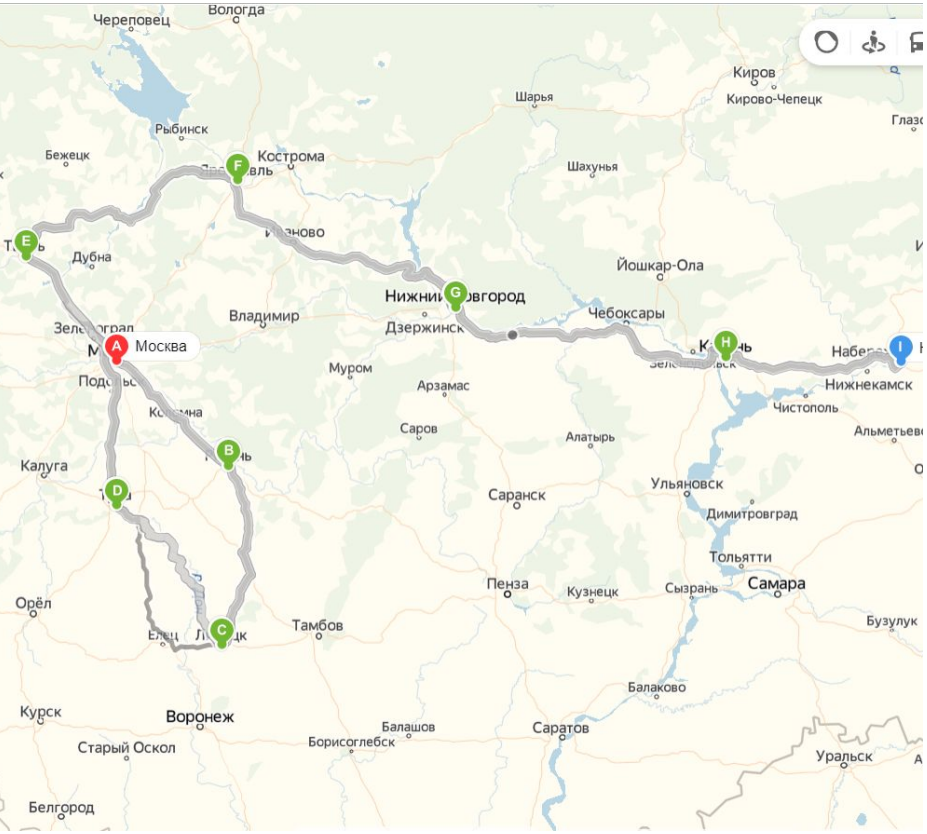
Сергиев Посад - Переславль-Залесский - Ростов Великий - Ярославль - Кострома - Иваново - Суздаль - Владимир - Сергиев Посад

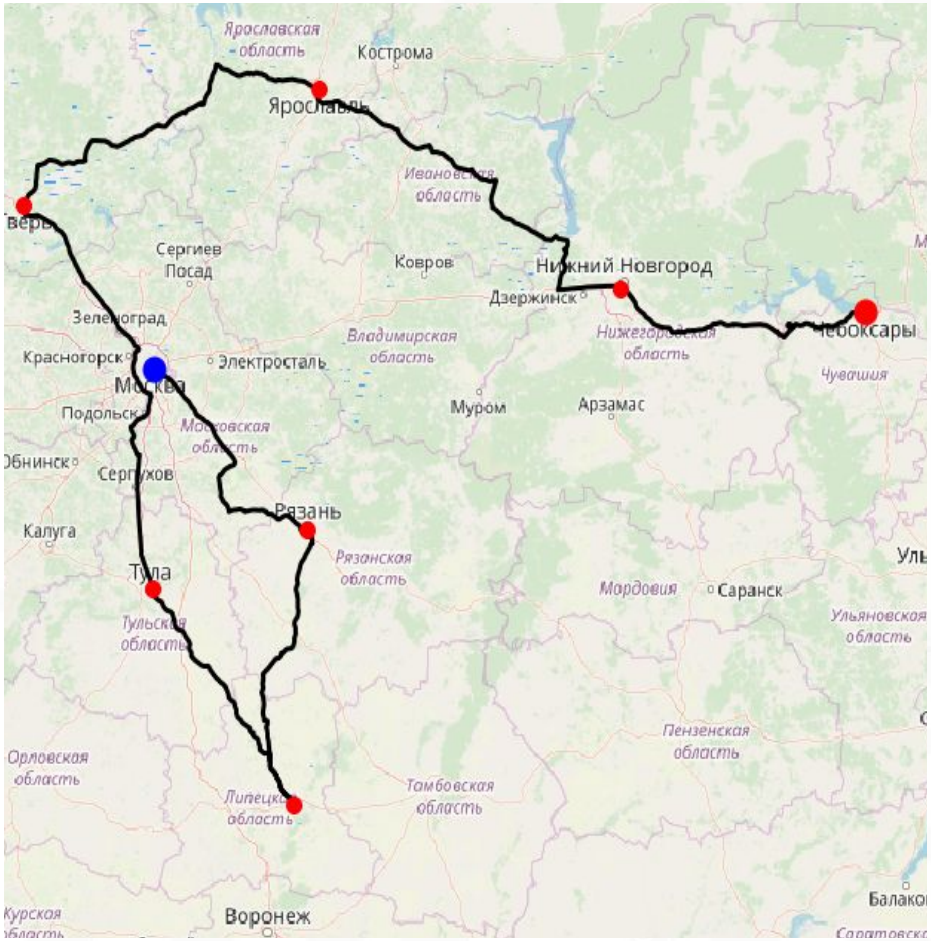
Москва - Арзамас



Москва - Рязань - Липецк - Воронеж - Белгород - Тула - Тверь - Ярославль - Нижний Новгород - Арзамас

Москва - Набережные Челны





Москва - Рязань - Липецк - Тула - Тверь - Ярославль - Нижний Новгород - Набережные Челны

Использованные ресурсы

Для выполнения проекта были использованы следующие ресурсы:

- “Route Planning in Road Networks” by Dominik Schultes
- Документация по NetworkX networkx.github.io/documentation/stable/
- Документация NumPy docs.scipy.org/doc/numpy
- Шейп-файлы с российскими дорогами были взяты с сайта diva-gis.org
- Изображения (тайлы) карт принадлежат сайту openstreetmap.org

Ниже приведена ссылка на репозиторий на сайте github.com с отчетом и .ipynb – файлом проекта.

https://github.com/egorakl/project_year2

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Спасибо за внимание!