

Тестовое задание для кандидата - Backend

- ◆ Использовать **Python 3** и фреймворк **Django**
- ◆ Для установки зависимостей предпочтительно использовать **pipenv**
- ◆ В качестве СУБД — **PostgreSQL** с расширением **PostGIS** (для хранения геометрии)

В ходе решения задания поощряется использование математических алгоритмов для решения поставленной задачи.

Задание

- 1) Необходимо создать Django приложение, в котором 2 модели: **Point** — узел графа и **Line** — связи узлов графа (схема данных на рисунке 1)
 - ◆ Поле **geom** — геометрический объект типа Point ([Django GEOS API](#))
 - ◆ Поля **from_point**, **to_point** : внешние ключи (**FK - ForeignKey**) на модель **Point**
 - ◆ **score** — кол-во баллов (дополнительный параметр каждой точки)

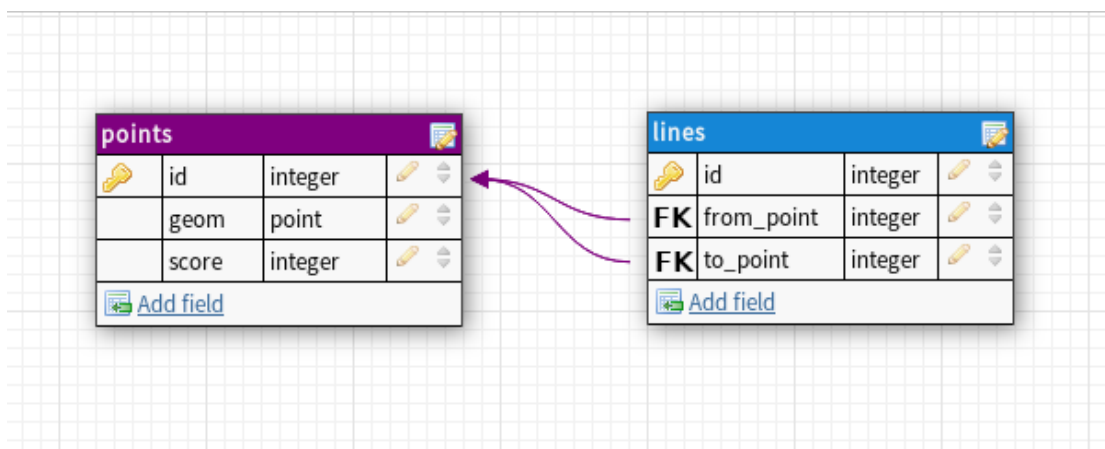


Рисунок 1 — Схема данных

- 2) Заполнить БД данными, полученными при обращении к публичному внешнему API: <https://datum-test-task.firebaseio.com/api/lines-points.json>, долготу и широту из файла преобразовать в тип *geometry* **PostGIS**, записывая в поле **geom**.

Для наглядности, данные из файла можно визуализировать следующим образом (рисунок 2). Красные числа — баллы, черные числа — номера точек.

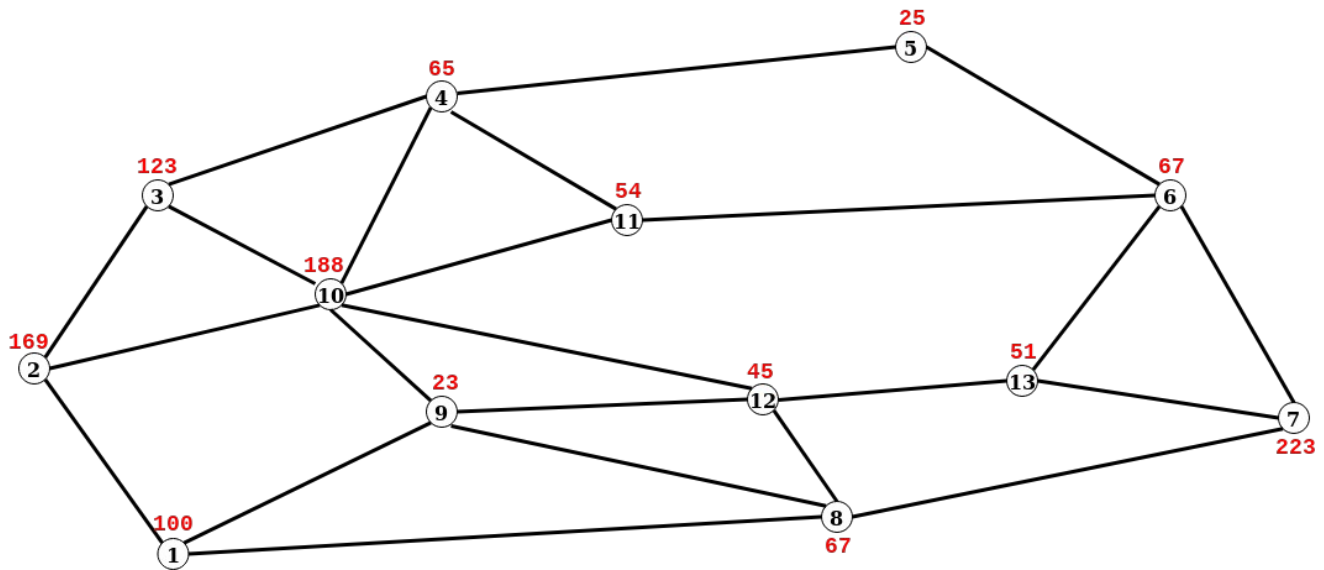


Рисунок 2 — Схема распределение баллов

4) Создать REST API используя Django rest framework для решения следующих задач:

- Поиск самой короткой линии, результат представить в км;
- Поиск линии имеющей минимальное количество баллов;

В GET-запросе передаваемые параметры — это начальная и конечная точка.

Как один из вариантов, с использованием [@action декоратора](#):

`localhost/api/points/1/min_score/?to=5`

`localhost/api/points/1/min_length/?to=5`

На рисунке 3 прямая от точки 2 до точки 6, суммарно набрал $169 + 188 + 54 + 67 = 478$ баллов, а длина равна ~4.7 км.

Ответом на GET-запрос должен быть GeoJSON.

Результаты вычислений не обязательно хранить в БД, приветствуется использование [django cache](#).

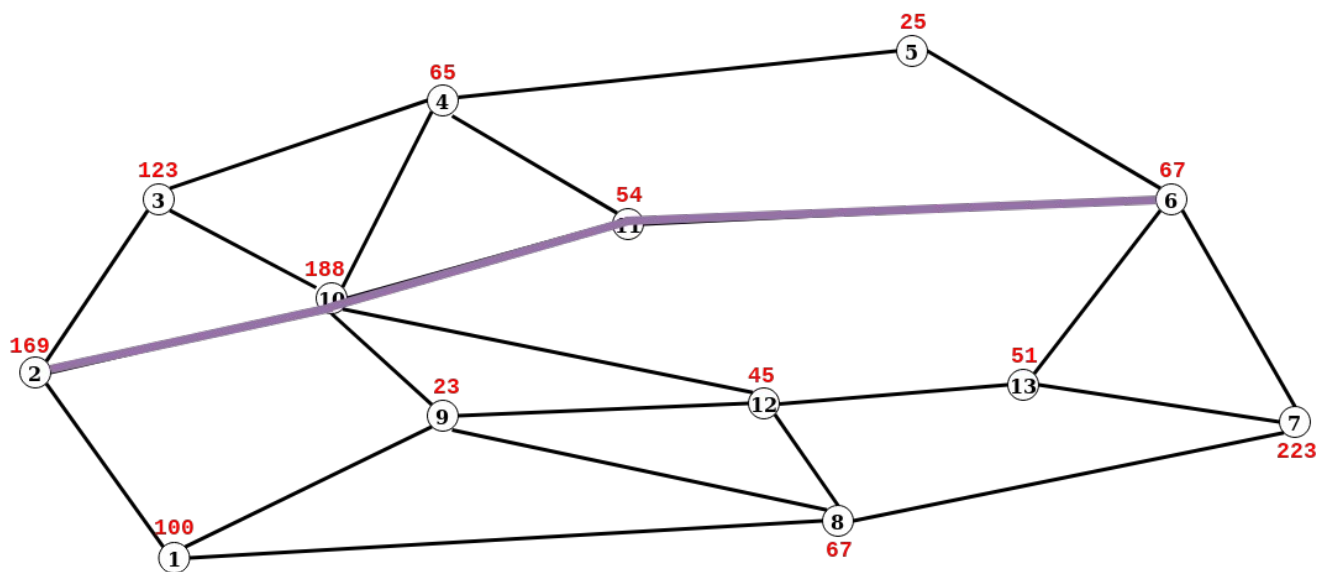


Рисунок 3 — Схема построения линии

5) Добавить стандартную django-админ панель, зарегистрировать модели со следующими возможностями:

- создание записей;
- удаление;
- обновление;

Важно, чтобы для поля **geom** было визуальное отображение точек и линий на карте, как на рисунке 4.



Рисунок 4 — Геометрия в админ панели

Задание на дополнительные баллы

- 1) Написание тестов для приложения и создание фикстур для БД. Тестовые данные должны загружаться в БД с помощью команды `python manage.py loaddata`
- 2) Используя API Yandex геокодера, определить физические адреса точек на карте. Расширить модель Point текстовым полем для адреса. Геокодировать геометрию при GET-запросе к конкретной точке. Записывать в БД или кэшировать (django cache).

Куда отправлять результаты тестового задания

Результат предоставить на почту: develop47@datum-group.ru (Андрей) ссылкой на Git репозиторий.