

The goal of this exercise sheet is to implement the geocoder. We have provided a few optional tasks, but you are free to add features as you like. These features should be documented and be used as the main part of the final presentation.

## Mandatory Tasks

### Task 1: String Preprocessing

It can happen that there are multiple valid ways to write names that are part of the address. For example streets can be written as Strasse, Straße or Str. This and other things like  $\acute{s}$  or casesensitivity might lead to problems in the query. Therefore, think about situations where this might arise and try to standardize them.

### Task 2: Geocoder

A Geocoder receives as input a string (e.g. “Hauptstrasse 10 Aalen”) and returns one or multiple objects described by the input. You can assume here that there are no spelling errors in the input string. There are at most minor things like mentioned in the previous task.

As a base routine implement something like an inverted index and fill it with the names of streets and shops, restaurants, etc. Querying this would, for example, result in every Hauptstrasse in the data set. Then you can use a secondary data structure to find the ones in Aalen. Return the results as a list as well as showing them on the map.

One challenge here is the spacing in the input string. As an example consider the input “Oberer Grundweg Vaihingen”, then it may be not clear if Oberer is a city and Grundweg Vaihingen is a street or the other way around.

Please add a text input field to the GUI to enter the queries and add timings for the queries.

### Task 3: Heuristics

A query might result in multiple results. Think about a way to order them in your results. Also the queries are given in natural language. How do you interpret them? Document your choices, including your reasoning, in a readme file.

### Task 4: Timings and other measurements

If not already done, measure the time it takes for each part of your code. This includes reading the pbf-file, all your point in polygon tests and queries to the geocoder and the reverse geocoder. Additionally log the number of buildings, streets, and regions you have in your data set.

Add an option to save and load your data from/into a binary file.

## Optional Tasks

Here is a nonexhaustive list of optional tasks you can implement to improve your geocoder. You can also come up with your own feature ideas.

### Task 1: Substring search

So far only exact matches in the string search will be found. Replace the inverted index with a suffix array or tree, so also partial matches can be found.

### Task 2: Aggregating

There might be many results close to each other. Bundle these.

### Task 3: Results in View

Only show results in the current view. You can combine this with the heuristics from the mandatory task. What if there are no results in the current view?

### Task 4: Example Queries

In the following are some example queries in roughly increasing difficulty. Make it possible to answer these queries.

- “Aalen Bahnhofstrasse 10”
- “Stuttgart Burger King”
- “Closest Park to Kaistrasse 5, Kiel”
- ...