

## Mandatory Tasks

### Task 1: Fast Point in Polygon Test

For the last step of the preprocessing, an efficient implementation of a Point in Polygon Test (PIP) is needed. We want to know in which city/state/zip-code/... all houses are located. As a base routine to determine that we need a point-in-polygon test. Be aware the latitude and longitude are not coordinates on a plane.

**Some of the following Resources might help**

- Many calculations for lat long points

### Task 2: Save Attributes to Houses

Find all administrative regions in which a house lays and save that information for the houses.

### Task 3: Reverse Geocoder

A reverse Geocoder receives as input a point on the map and returns the closest object to that point. Along with the closest object, its full address should be returned. Implement that feature by using a data structure of your choice (Grid/Quadtrees/k-d tree/...), but **not** by only iterating over an array.

### Task 4: Front end

Your front end should work with the reverse Geocoder so it should be possible to set the coordinates by clicking on the map, and the results should be visualized. It should be able to visualize points (houses), polylines (streets) and polygons (cities/...).

## Optional Tasks

### Task 1: Objects in View

Find all objects in the current view. Again, think about a suitable data structure.

### Task 2: Reverse Geocoder for all the other things

So far the reverse Geocoder only returns houses. In some situations it might make sense to return other objects like cities, countries, ... For example think about the following questions:

What happens if there is no close object?

How and when should an area/city/... be returned?

How can the distance to a polygon be determined/approximated?