

Closest city problem

The closest pair problem is a “classic” algorithm for beginning coders. It is a problem of computational geometry: given n points in metric space, find a pair of points with the smallest distance between them. It has real life significance in many applications, especially maps. Consider: where are the closest shops? where is the closest hospital?

TASK ONE

Given two sets of coordinates, use the haversine formula to determine the distance between them.

The haversine formula determines the great-circle distance between two points on a sphere given their longitudes and latitudes. Where

- d = the distance between the two points (along the surface of the sphere),
- r = the radius of the sphere,
- φ_1 = latitude of point 1
- φ_2 = latitude of point 2,
- λ_1 = longitude of point 1,
- λ_2 = longitude of point 2.

Then, the distance between those two points is

$$= 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

NOTE

- You may assume the radius of the earth is 6373 kilometers.
- You will likely need to import your Math library.
- Most programming languages use radians by default so you will need to convert your degree based coordinates to radians using the appropriate function in your Math library.

TASK TWO

- Download the cities.csv file from <https://github.com/paulbaumgarten/data-sets>
- Load all the coordinates from the CSV file into your program
- Prompt the user for two city names
- Find the distance between the two given cities

TASK THREE

Given an input of one city name, find the closest other city to it from the data in the CSV file in Task Two.