Computational Geometry

Labwork 5: Solving the kNN problem

November 25, 2020

The 1NN problem can be solved efficiently with a k-d tree data structure.

- The method is described in `ADS-L08a.pdf`
- A java implementation of the static method

  `double[] kdTree.NN1(kdTree kd,double[] X)`

  of class `kdTree`, which returns an array with the coordinates of the point which is nearest to `X` in a k-d tree `kd`.

You are asked to extend the implementation of `kdTree` with the static method

`BPQ<double[]> kdTree.NNk(kdTree kd,int k,double[] X)`

which returns the `k` nearest neighbors to the test point `X` in the k-d tree `kd`. This method can be obtained by changing method `NN1` as follows: instead of keeping track only of a globally visible best guess

```
double[] guess;
```

we keep track of k best guesses which can be stored in a globally visible bounded priority queue

```
BPQ<double[]> guesses;
```

The current implementation of class `kdTree` has a dummy implementation of the required method:

```
public static BPQ<double[]> NNk(kdTree kd, int k, double[] test) {
  // TODO
  return null;
}
```

Complete this implementation to work as expected.