Algorithms and Computational Thinking
Autumn 2017

Tuesday, 14th November 2017

Exercise 1 - Spatial Tree Algorithms

k-d Tree

A *k-dimensional* tree is a data structure to organize multiple points in a *k-dimensional* space. The algorithm to build a *k-d* tree is explained on the following link : `https://en.wikipedia.org/wiki/K-d_tree` (wikipedia).

By following the pseudo-code described below, the goal of the first part of the exercise is to implement a *k-d* tree. Then, the goal of the second part of the exercise is to implement a search function by following the nearest neighbour search algorithm also detailed in the same wikipedia page. It is mandatory to do the exercise in Python, Scala and Swift. You must consider the following set of points to build the tree : (2,3), (5,4), (9,6), (4,7), (8,1), (7,2).

k-d Tree - Creation function

```
function kdtree (list of points pointList, int depth)
{
    // Select axis based on depth so that axis cycles through all valid values
    var int axis := depth mod k;

    // Sort point list and choose median as pivot element
    select median by axis from pointList;

    // Create node and construct subtrees
    var tree_node node;
    node.location := median;
    node.leftChild := kdtree(points in pointList before median, depth+1);
    node.rightChild := kdtree(points in pointList after median, depth+1);
    return node;
}
```

Source : wikipedia.