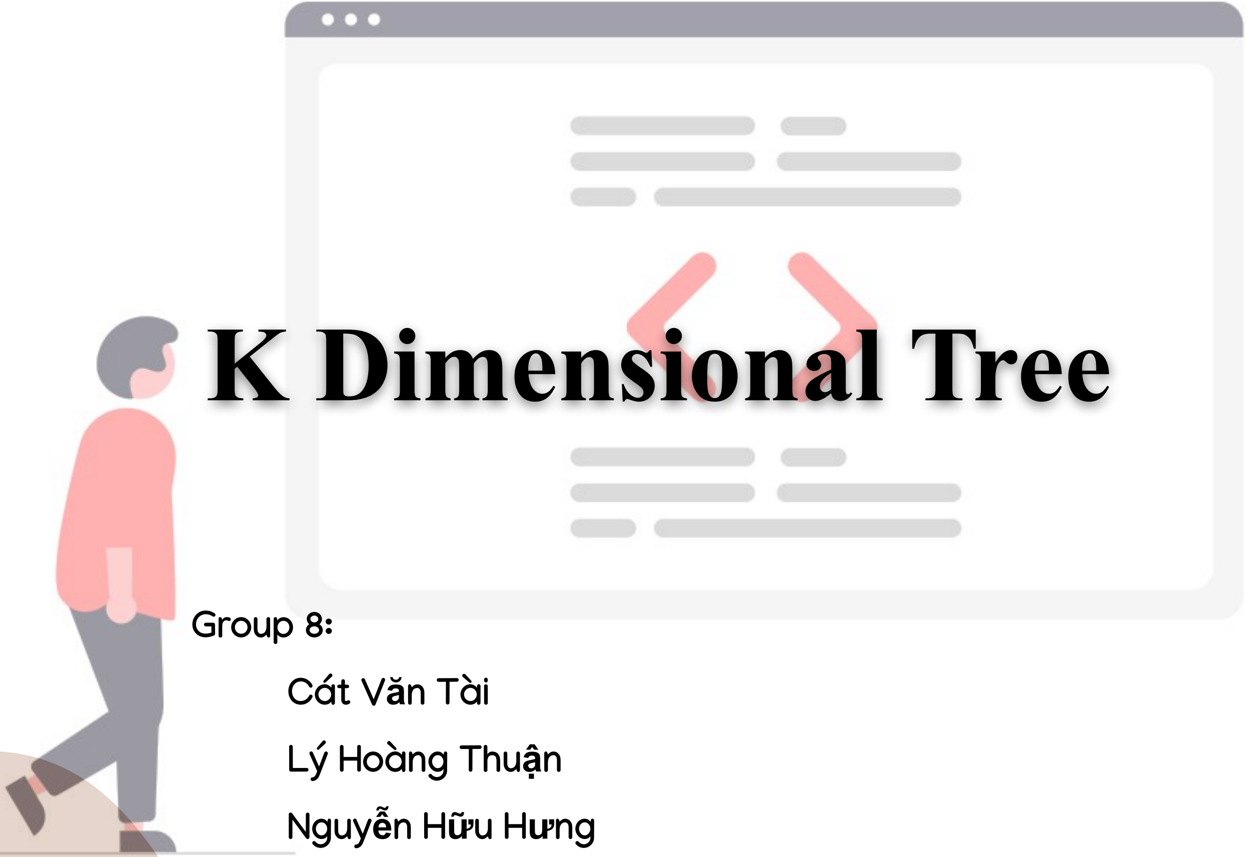


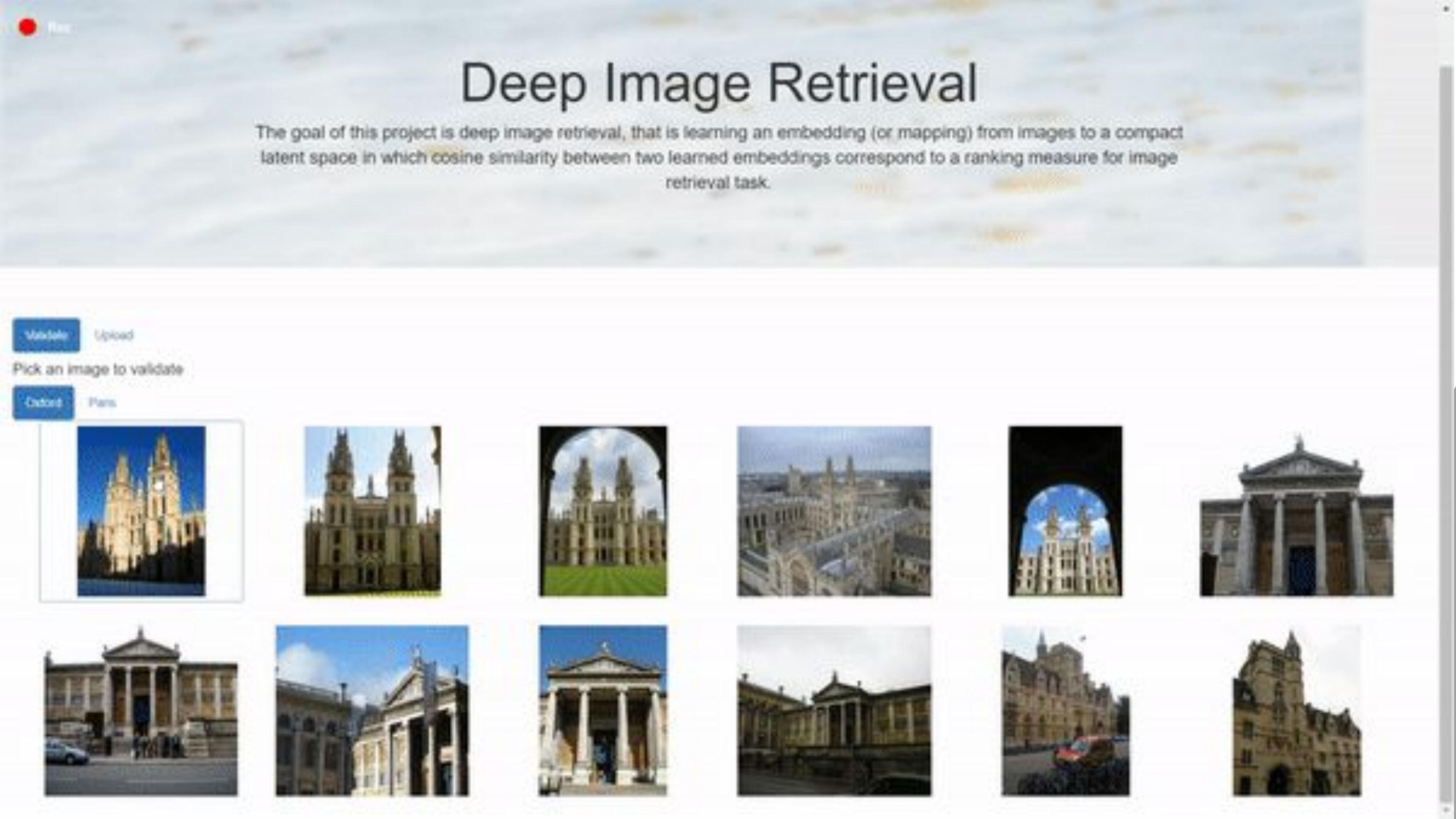
K Dimensional Tree

Group 8:

Cát Văn Tài

Lý Hoàng Thuận

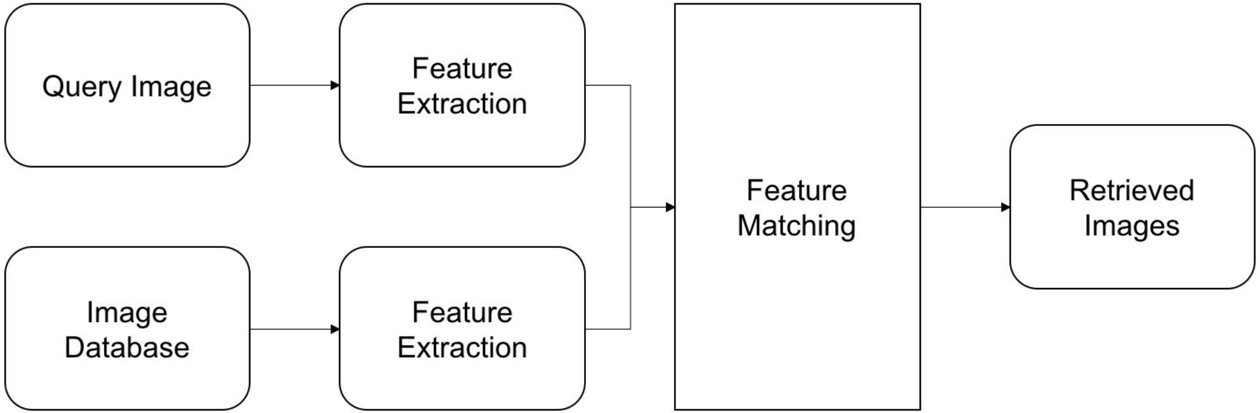
Nguyễn Hữu Hưng



Deep Image Retrieval

The goal of this project is deep image retrieval, that is learning an embedding (or mapping) from images to a compact latent space in which cosine similarity between two learned embeddings correspond to a ranking measure for image retrieval task.

introduce IR

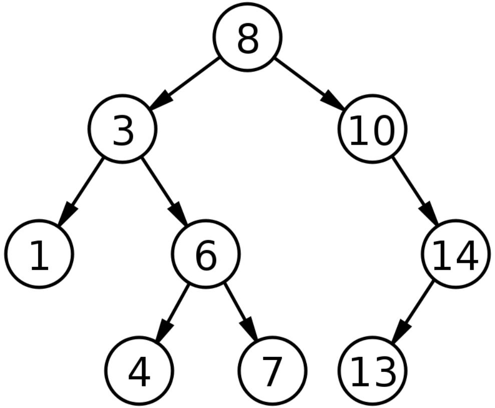


The pipeline for IR

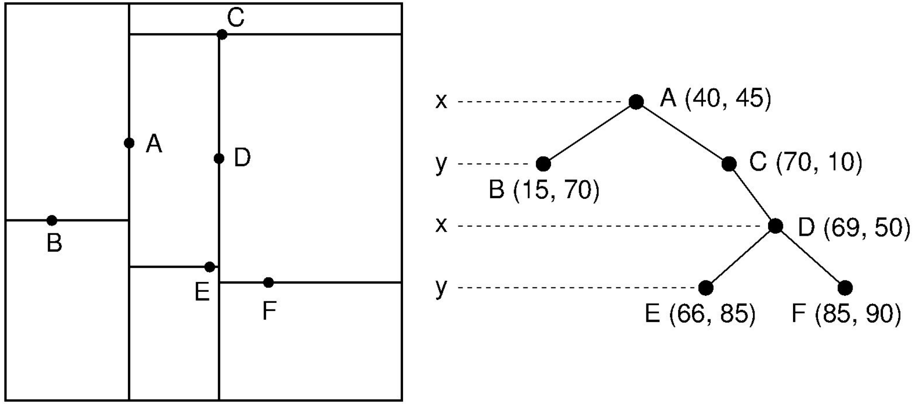
- Given 100.000 samples with n_features is 128. Then, input a any sample and convert it to vector 128. Let's find the nearest samples compare with the input.
- Solutions for feature matching: linear search, k-d tree, etc.

k-d tree

- A k-d tree is a binary search tree for partitioning data
- It is used for various applications like nearest points, range search



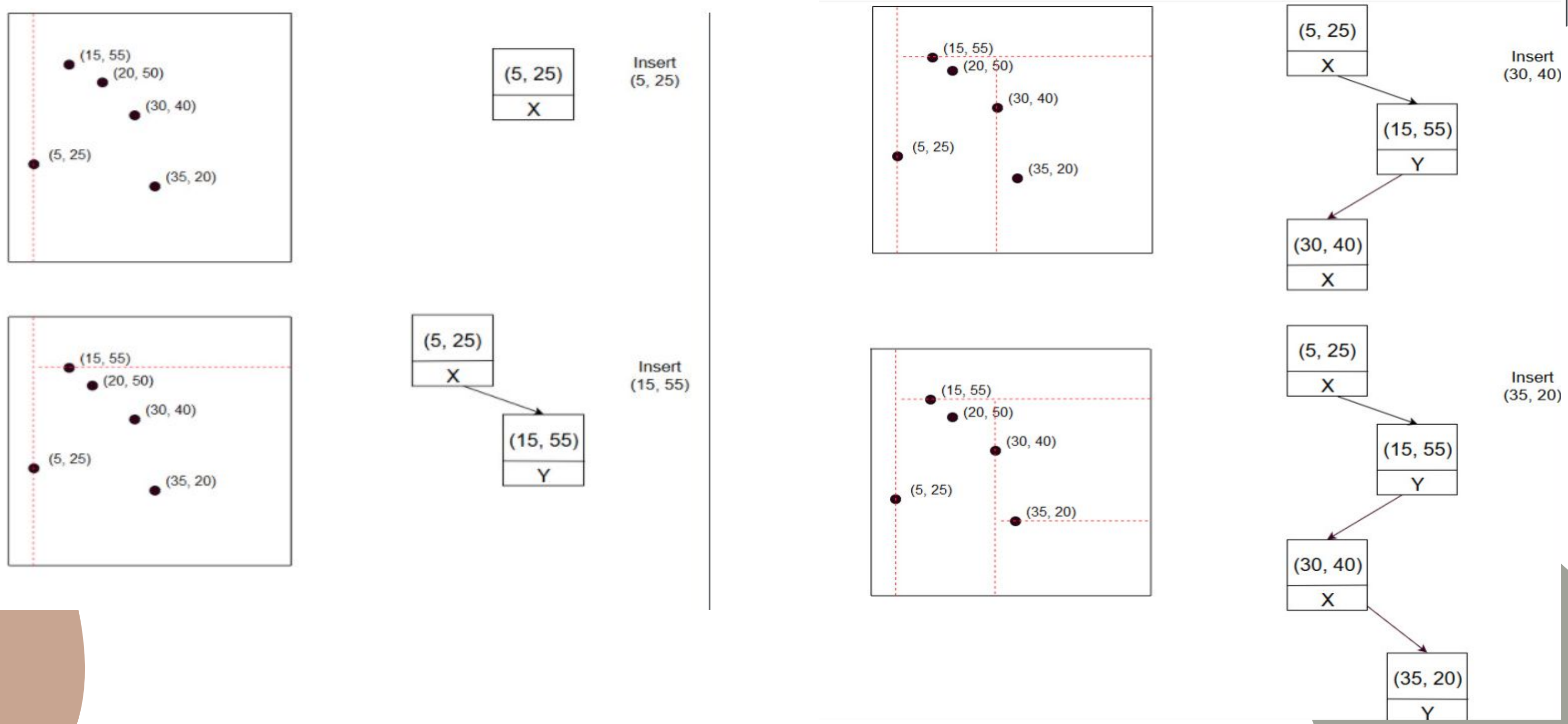
The representation of binary tree



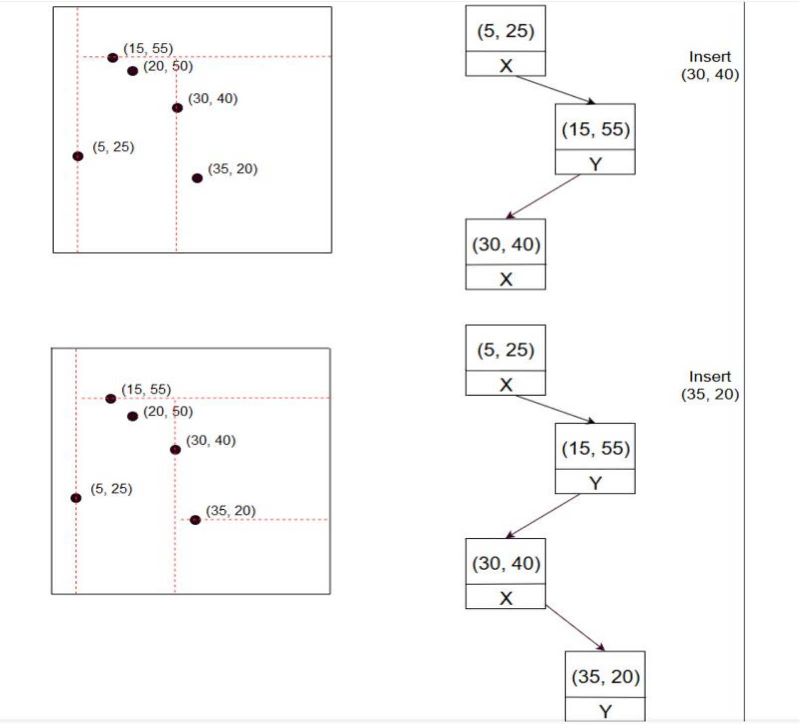
The representation of k-d tree

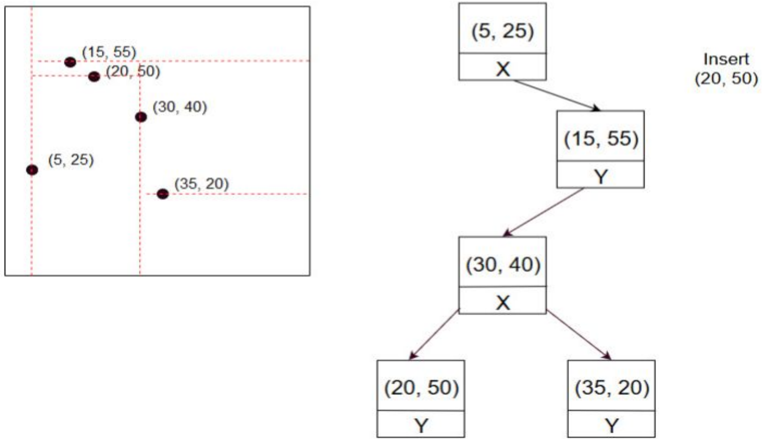
construction

There are many possible ways to choose axis-aligned splitting planes, so there are many different ways to construct k-d trees.

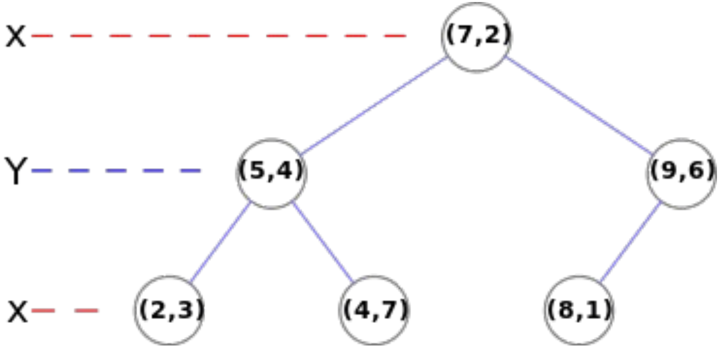


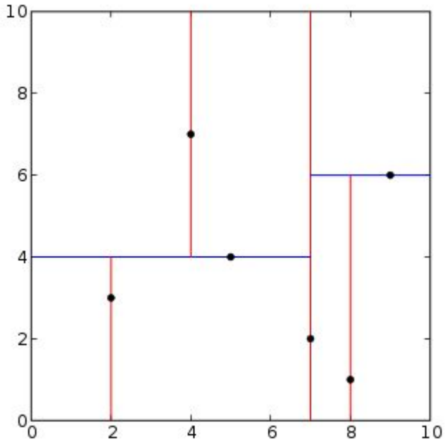
construction





construction



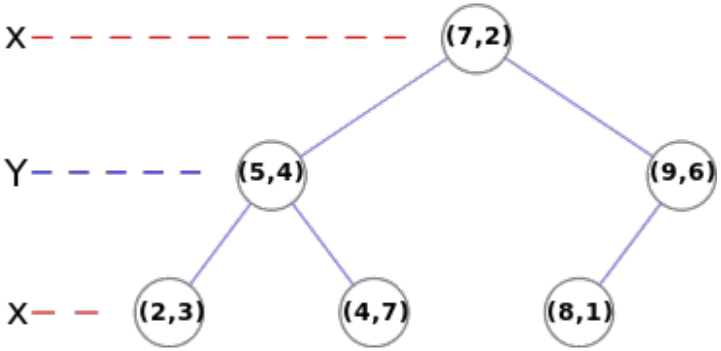


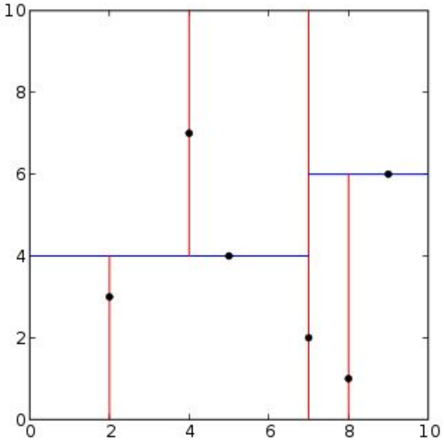
The canonical method of k-d tree construction has the following constraints:

- As one moves down the tree, one cycles through the axes used to select the splitting planes
- Points are inserted by selecting the median of the points being put into the subtree

A popular practice is to sort a fixed number of randomly selected points, and use the median of those points to serve as the splitting plane. In practice, this technique often results in nicely balanced trees.

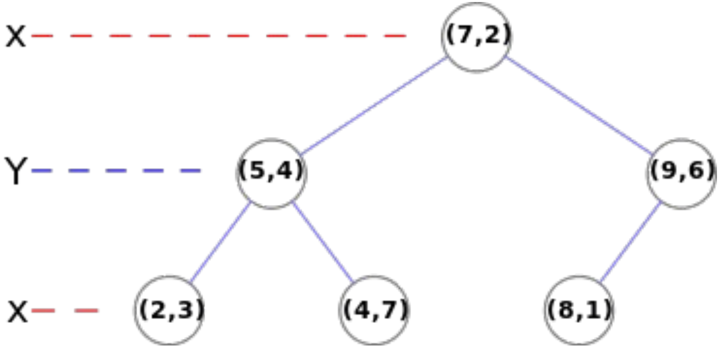
search

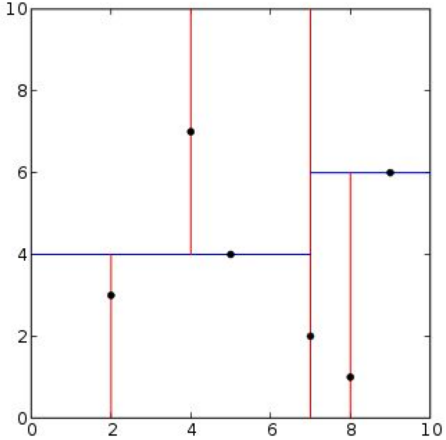




- Nearest neighbour search
- K nearest neighbours search
- Approximate nearest neighbour search
- Range search

adding elements





- First, traverse the tree, starting from the root and moving to either the left or the right child depending on whether the point to be inserted is on the "left" or "right" side of the splitting plane.
- Once you get to the node under which the child should be located, add the new point as either the left or right child of the leaf node, again depending on which side of the node's splitting plane contains the new node.

Adding points in this manner can cause the tree to become unbalanced, leading to decreased tree performance

experiment

- Given 100.000 samples with n_features is 128. Then, input a any sample and convert it to vector 128. Let's find the nearest sample compare with the input.
- Solutions: linear search, k-d tree, etc.

Solution	Time (s)	Accuracy(%)
linear search	0.44	100
k-d tree	0.02	100

Table comparison results between linear-search and k-d tree

Algorithm	Average	Worst case
space	O(n)	O(n)
search	O(logn)	O(n)
insert	O(logn)	O(n)
delete	O(logn)	O(n)

Time complexity k-d tree in big O notation

homeworks

- Build k-d tree (build, nearest neighbour search, range search) from scratch
- Find out balancing, removing elements operators (optional)

Thanks for listening