Construction functions

CoverTree.from_matrix(*points***)**

Constructs a new cover tree on a given collection of points. The function utilizes multiple cores.

Parameters:	**point***: 2D ndarray* A contiguous numpy array of shape (num_ points, dim)..
Returns:	**Cover_tree***: CoverTree* Returns an object of class CoverTree containing the points arranged in the tree.

Manipulation functions

CoverTree.insert(*point***)**

Inserts a new point into the cover tree. [Thread-safe]

Parameters:	**point***: 1D ndarray* A contiguous numpy array of same dimension as other points.
Returns:	**success***: Boolean* Returns true is the point was successfully inserted. If the point to be inserted is a duplicate one, i.e. already present in the tree then it is not inserted and a false is returned.

CoverTree.remove(*point***)**

Removes a point from the cover tree. [NOT Thread-safe]

Parameters:	**point***: 1D ndarray* A contiguous numpy array of same dimension as other points.
Returns:	**success***: Boolean* Returns true is the point was successfully removed. If the point to be removed is not found in the tree, then a false is returned. Also if the point to be removed is the root of the tree a false is returned with sorry message that efficient way to delete root node is not known.

Search functions

CoverTree.NearestNeighbour(*points***)**

Searches for the nearest neighbour for each point in the cover tree. [Thread-safe]

Parameters:	**points***: 2D ndarray* A contiguous numpy array of shape (num_query_points, dim).
Returns:	**neighbours***: 2D ndarray* Returns the nearest neighbour for each of the point. Same shape as points.

CoverTree.kNearestNeighbour(*points, k=10***)**

Searches for the k nearest neighbours for each point in the cover tree. [Thread-safe]

Parameters:	**points***: 2D ndarray* A contiguous numpy array of shape (num_query_points, dim).
Returns:	**neighbours***: 3D ndarray* Returns the k nearest neighbours for each of the point. Shape is (num_query_points, k, dim)

Auxiliary/Test functions

CoverTree.display(*points, k=10***)**

A poor quality diagram of tree structure is printed [Use with small number of points/dims]

Parameters:	-
Returns:	**-**

CoverTree.test_covering(*points, k=10***)**

Checks for the covering property of the data structure over all nodes

Parameters:	**-**
Returns:	**-**