

USE AVL_TREE FOR BFS_TRAVERSAL

What we want the code to do : BFS_Traversal (root, left, right)

What do we need to put in : (avl_tree)

Pseudo code : BFS_traversal ()

BFS_helper(subroot)

Print root;

BFS_helper(root->left)

BFS_helper(root->right);

./

NEED TO GENERATE A GRAPH FOR THESE ALGORITHMS

Figure out how to create a graph class

Each article is a node

Each link to a new article is an edge

Dog -> Husky

^

|

|

Corgi

Node-Edge-Node

|

Edge

|

Node

What we want the code to do : LandMark_Path_Algorithm Node A - -- c ----- B

What do we need to put in : (Node A, Node B, Node C)

Pseudo code : LMP(key A, key B)

path1 = ShortestPath(A, C)

path2 = ShortestPath(C, B)

Path = path1 +path2

return Path

Animal to Husky??

Animal -> Dog -> Husky

Path(Animal, dog) + Path(Dog, Husky)

What we want the code to do : Dijkstra's Algorithm

What do we need to put in :(Node A, Node B)

Return shortestPath(A,B)

Pseudo code :

Helper functions :

Shortest path

BFS_helper

Work that needs to be done :

-Figure out how to read from file

-Figure out how to create graph class

-complete helper functions

-compile it correctly (in case there are errors in makefile and stuff)