
Implement the following problems for Graphs

Q1. Write a program to find approachable nodes from a given source of a given graph using queue as an intermediate data structure (BFS).

Q2. Write a program to traverse various nodes of a given graph using stack as an intermediate data structure (DFS).

Q3. Write a program to find shortest path from a given source to all the approachable nodes in a weighted graph

Q4. Implement the full AVL tree inserts function, and writes a program that demonstrates that your AVL tree is capable of handling a number of insertions while remaining balanced.

Q5. Given two AVL trees T1 and T2, where the largest key in T1 is less than the smallest key in T2, Join(T1, T2) returns an AVL tree containing the union of the elements in T1 and T2. Give an algorithm (in pseudo code) for Join that runs in time $O(\log n)$, where n is the size of the resulting AVL tree. Justify the correctness and efficiency of your algorithm.

Q6. Write a program to check whether a given graph represents a ring Topology

Q7. Write a program to check whether removing a edge makes a graph disconnected.