BFS: From the beginning node, put all neighbor nodes into FIFO queue, then mark these neighbor nodes’ distance between the beginning node is 1, then mark the beginning node as visited, which won’t be visited again any more. After, pop the first in node from queue, repeat the above procures, until the whole queue is empty.

DFS: From the beginning node, recursively visit all neighbor nodes, then visit all neighbor nodes’ neighbor. For example, if node A is the first neighbor of beginning node, and node B is a neighbor of A, then visit B after A. If there are unvisited neighbor of B, then visit these neighbors of B, otherwise continue visit unvisited neighbors of A. After all neighbors of A are visited, then start from other unvisited node except A.

Difference: If the tree is very deep and solutions are rare, DFS might take an extremely long time, but BFS could be faster. If the tree is very wide, a BFS might need too much memory, so it might be completely impractical. If solutions are frequent but located deep in the tree, BFS could be impractical. If the search tree is very deep you will need to restrict the search depth for DFS, anyway (for example with iterative deepening).