**Question**

Program to implement best first search and depth first search.

**Description**

Best First Search (BFS):

Best First Search (BFS) is a type of search algorithm used for traversing or searching tree or graph data structures. It aims to find the most promising path to the goal based on a heuristic function. BFS evaluates which node to explore next based on an evaluation function, typically using a priority queue to maintain the order of node exploration.

Depth First Search (DFS):

Depth First Search (DFS) is a fundamental algorithm for traversing or searching tree or graph structures. Unlike BFS, which expands nodes level by level, DFS explores as far down a branch as possible before backtracking. DFS is often used to explore all possible paths or to check if there exists a path between two nodes.

**Programming Implementation**

* BFS implementation:

vector<vector<int>> levelOrder(TreeNode\* root) {

        vector<vector<int>> levelorder;

        if(root == nullptr) return levelorder;

        queue<TreeNode\*> q;

        q.push(root);

        while(!q.empty()){

            int size = q.size();

            vector<int> level;

            for(int i=0; i<size; i++){

                TreeNode\* node = q.front();

                q.pop();

                level.push\_back(node->val);

                if( node->left != nullptr) q.push(node->left);

                if(node-> right != nullptr) q.push(node->right);

            }

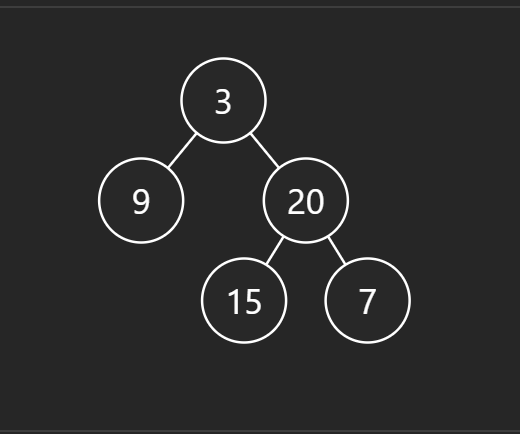
            levelorder.push\_back(level);

        }

        return levelorder;

    }

Above is a function that will take a binary tree as input and it will return the BFS traversal of that tree.

Input:

[3,9,20,null,null,15,7]

Output:

[[3],[9,20],[15,7]]

* DFS implementation:

void preorder(TreeNode\* root, vector<int>& ans){

        if (root == NULL) return ;

        ans.push\_back(root->val);

        preorder(root->left, ans);

        preorder(root->right, ans);

    }

    vector<int> preorderTraversal(TreeNode\* root) {

        vector<int> ans;

        preorder(root,ans);

        return ans;

    }

Above is a function that takes a binary tree as input and returns the DFS traversal of that tree.

A diagram of numbers and circles

Description automatically generatedInput:

[1,2,3,4,5,null,8,null,null,6,7,9]

Output:

[1,2,4,5,6,7,3,8,9]