1530. Number of Good Leaf Nodes Pairs

Root approach

```
/*
   Preprocess all paths from root
   For each pair of paths:
   ---compute the distance between the pair of leafs
   ---increment the answer if the distance <= given distance

   Time complexity: O(n+(nb_leafs^2)*n)=O(n^3)
   Extra space complexity: O(2n)=O(n)
*/
class Solution {
   public:
      std::string path;
      std::vector<std::string> paths;
   public:
      void get_all_paths_to_leafs_from_root(TreeNode* root){
         if(!root) return;

         path.push_back('L');
         get_all_paths_to_leafs_from_root(root->left);
         path.pop_back();

         path.push_back('R');
         get_all_paths_to_leafs_from_root(root->right);
         path.pop_back();

         if(!root->left&&!root->right) paths.push_back(path);
      }
```

```
int get_distance_between_two_leafs_from_paths_from_root(std::string& path1, std::string& path2){

        int i=0,j=0,n=path1.size(),m=path2.size();
        while(i<n && j<m && path1[i]==path2[j]){
            i++;
            j++;
        }
        return n+m-2*i;

    }

    int countPairs(TreeNode* root, int distance) {
        get_all_paths_to_leafs_from_root(root);
        int nb_leafs=paths.size();
        int ans=0;
        for(int i=0;i<nb_leafs-1;++i){
            for(int j=i+1;j<nb_leafs;++j){
                int dist=get_distance_between_two_leafs_from_paths_from_root(paths[i],paths[j]);
                if(dist<=distance) ans++;
            }
        }
        return ans;
    }
};
// @lc code=end
```