

2096. Step-By-Step Directions From a Binary Tree Node to Another

```
/*
    Recursion,backtracking,DFS,two pointers
    Time complexity: O(n)
    Extra space complexity: O(n)
*/
class Solution {
public:
    void find_path(TreeNode* p,int val,std::string& path_tmp,std::string& path){
        if(!p) return;

        if(p->val==val) path=path_tmp;

        path_tmp.push_back('L');
        find_path(p->left,val,path_tmp,path);
        path_tmp.pop_back();

        path_tmp.push_back('R');
        find_path(p->right,val,path_tmp,path);
        path_tmp.pop_back();
    }

    string getDirections(TreeNode* root, int startValue, int destValue){
        std::string tmp,path_start;
        find_path(root,startValue,tmp,path_start);

        tmp="";
        std::string path_dest;
        find_path(root,destValue,tmp,path_dest);

        int i=0,j=0;
        while(i<path_start.size()&&j<path_dest.size()&&path_start[i]==path_dest[j]){
            i++;
            j++;
        }

        /*std::cout<<path_start<<"/"<<path_dest<<"\n";
        std::cout<<i<<"/"<<j<<"\n";*/

        path_start=path_start.substr(i,path_start.size()-i);
        path_dest=path_dest.substr(i,path_dest.size()-i);

        for(auto& c: path_start) c='U';

        return path_start+path_dest;
    }
};
```