**Number of balanced Binary Trees**

Given an integer h, find the possible number of balanced binary trees of height h. You just need to return the count of possible binary trees which are balanced.

**This number can be huge, so return output modulus 10^9 + 7.**

Write a simple recursive solution.

#include<cmath>

using namespace std;

int balancedBTs1(int h,int b[]){

int mod = (int)pow(10,9) + 7;

if(h==0){

b[0]=1;

return b[0];

}

if(h == 1){

b[1]=1;

return b[1];

}

if(b[h]!=0){

return b[h];

}

else{

int x = balancedBTs1(h-1,b);

int y = balancedBTs1(h-2,b);

int firstValue = (int)(((long long)x\*x)%mod);

int secondValue = (int)(((long long)(x) \* y \* 2)%mod);

int ans = (int)(((long long)(firstValue) + secondValue)%mod);

b[h]=ans;

return b[h];

}

}

int balancedBTs(int h){

int b[h+1];

for(int i=0;i<=h;i++)

b[i]=0;

int k=balancedBTs1(h,b);

return k;

}