LeetCode

Number of Ways to Reorder Array to Get Same BST

class Solution {

vector<vector<long long>> dp;

long long MOD = 1e9 + 7;

unsigned long long solve(vector<int> &nums) {

if (nums.size() <= 1) return 1;

vector<int> l, r;

for (int i = 1; i < nums.size(); ++i) {

if (nums[i] > nums[0]) r.push\_back(nums[i]);

else l.push\_back(nums[i]);

}

int n = l.size(), m = r.size();

return solve(l) \* solve(r) % MOD \* dp[n + m][n] % MOD;

}

public:

int numOfWays(vector<int> &nums) {

dp = vector<vector<long long>>(nums.size() + 1, vector<long long>(nums.size() + 1, 0));

for (int i = 1; i < nums.size() + 1; ++i) {

dp[i][0] = 1;

dp[i][1] = i;

dp[i][i - 1] = i;

dp[i][i] = 1;

}

for (int i = 2; i < nums.size() + 1; ++i) {

for (int j = 2; j < nums.size() + 1; ++j) {

if (i >= j) dp[i][j] = (dp[i - 1][j - 1] % MOD + dp[i - 1][j] % MOD) % MOD;

else break;

}

}

return solve(nums) - 1;

}

};