//1번 피보나치 수열 문제 sol -> c++

Can’t sol -> js -> number 한계

#include <iostream>

using namespace std;

unsigned long long int arr[101];

unsigned long long int fibo(int n){

if(n <= 1) return n;

if(arr[n] != 0) return arr[n];

return arr[n] = fibo(n - 1) + fibo(n - 2);

}

int main(){

int n;

arr[1] = 0;

arr[2] = 1;

arr[3] = 1;

cin >> n;

cout << fibo(n + 1);

return 0;

}

//2번 배낭문제 sol -> js

function solution2\_1\_sol(){

let max = gets().trim();

max = Number(max);

let count = gets().trim();

count = Number(count);

let d = Array.from(Array(count + 1), () => Array(max + 1).fill(0));

let weight = gets().split(' ');

let val = gets().split(' ');

for(let i = 0; i < count; i++){

weight[i] = Number(weight[i]);

val[i] = Number(val[i]);

}

weight.unshift('.');

val.unshift('.');

for(let i = 1; i <= count; i++){

for(let j = 1; j <= max; j++){

d[i][j] = d[i - 1][j];

if(j - weight[i] >= 0){

d[i][j] = Math.max(d[i][j], d[i - 1][j - weight[i]] + val[i]);

}

}

}

print(d[count][max]);

}

//3번 최장공통부분수열 문제 sol -> js

function solution3\_1\_sol(){

let arr1 = gets().split('');

let arr2 = gets().split('');

arr1.unshift('1');

arr2.unshift('2');

let sub = Array.from(Array(arr1.length), () => Array(arr2.length).fill(0))

let max = 0;

for(let i = 1; i < arr1.length; i++){

for(let j = 1; j < arr2.length; j++){

if(arr1[i] == arr2[j]){

sub[i][j] = sub[i - 1][j - 1] + 1;

}else{

sub[i][j] = Math.max(sub[i - 1][j], sub[i][j - 1]);

}

}

}

max = sub[arr1.length - 1][arr2.length - 1];

print(max);

}