

Subtask-ul cu N=3 în valoare de 17 puncte

Pentru N=3 structura arborelui initial poate fi:

- Avem cele 2 muchii legate de nodul 1 (muchii 1-2 si 1-3). Numărul de noduri la distanța D de rădăcina va fi întotdeauna 2^D , din modul de exapandare al arborilor.
- Avem cele 2 muchii legate în lanț: 1-2 și 2-3 (ori, similar 1-3 si 3-2). Numărul de noduri la distanța D de rădăcină va fi întotdeauna 1, deoarece permanent lanțul doar se va lungi în jos.

Simularea efectiva a extinderii arborelui, soluție parțială în valoare de 26 - 30 puncte

O soluție de punctaj parțial ar fi să facem un DFS, unde să reținem permanent adâncimea la care se află nodul procesat. Astfel, atunci când suntem într-o frunză la adâncimea H, să continuăm DFS-ul cu nodul 1 și adâncimea H. Prin această procedură simulăm prelungirea arborilor doar atunci când avem nevoie. Vom da return din DFS doar atunci cand adâncimea actuala este egală cu adâncimea din cerință D, moment în care vom și incrementa rezultatul final.

Putem observa ca numarul total de noduri parcurse de aceasta abordare este $T[1] + T[2] + T[3] + \dots + T[D]$, unde $T[i]$ reprezinta numarul de noduri la distanta i de radacina 1. (practic, raspunsul la cerinta ar fi $T[D]$). Grosolan putem aproxima complexitatea la $O(\text{raspuns}^2)$ si de aceasta solutia va obtine punctajul doar pentru subtask-ul „raspuns $\leq 10\,000$ ” și puțin din subtask-ul N=3, mai exact cazul lanț.

Soluția de 90 de puncte

Soluția problemei constă în numărarea frunzelor în arborele initial.

Fie $FR[i]$ = numărul de frunze la distanța i de rădăcina 1 în arborele initial

Fie $NT[i]$ = numărul de noduri la distanța i de rădăcina 1 în arborele initial care **nu** sunt frunze.

Știm ca arborele initial are doar 100 de noduri, deci acești 2 vectori de mai sus FR și NT nu vor fi deloc mari.

În continuare vom defini $V[i]$ ca fiind numărul de noduri la distanța i care au fost cândva frunze în procesul de extindere al arborelui initial.

Initial $V[0] = 1$ și $V[i] = 0$ oricare $i \neq 0$ (caz particular de pornire. Dacă extindem un nod izolat folosind tehnica din enunt vom obtine exact arborele initial).

Putem parcurge de la 0 la D vectorul V pentru a genera frunze la dreapta în vector folosind tehnica programării dinamici. Mai exact, o recurenta înainte de forma:

fiind la poziția i și având deja calculata valoarea $V[i]$

$V[i + 1] += V[i] * FR[1]$

$V[i + 2] += V[i] * FR[2]$

.....

$V[i + \text{adancime_arbore_initial}] += V[i] * FR[\text{adancime_arbore_initial}]$

În cuvinte, dacă la distanța i avem $V[i]$ noduri și știm că în arborele initial aveam $FR[k]$ frunze la distanța k de rădăcina 1, atunci prelungind toate cele $V[i]$ noduri aflate la distanța i , vom obține $V[i] * FR[k]$ noi frunze la distanța $V[i] + k$.

Acum, pentru a afla exact câte noduri sunt la distanța D de rădăcina 1, trebuie să nu uităm de nodurile din arborele initial care nu sunt frunze, cele reținute în vectorul NT .

Răspunsul = $V[D] + V[D - 1] * NT[1] + V[D - 2] * NT[2] + \dots + V[D - a_a_i] * NT[a_a_i]$, unde a_a_i este adâncimea arborelui initial.

Clarificare pentru formula de mai sus:

$V[D]$ reprezintă numărul de noduri foste frunze aflate la distanța D de rădăcina (acestea clar trebuie numărate).

$V[D - 1] * NT[1]$ reprezintă nodurile-funze aflate la distanța $D-1$ de rădăcina, pe care le prelungim pentru ultima dată. Pentru fiecare arbore astfel prelungit trebuie să numărăm câte noduri nefrunze (deoarece frunzele au fost deja numărate) are la distanța 1 – informație reținută în $NT[1]$.

$V[D - k] * NT[k]$ exact ca mai sus.

Complexitate timp finală $O(\text{MAX_D} * \text{MAX_N})$, adică $D * \text{adancime_arbore_initial}$

Aveți grijă permanent să nu uitați de modulo.