

## Double Agents (trees)


De Britse spionagedienst MI6 is aan het plannen om de criminele organisatie SPECTRE te infiltreren met een netwerk van dubbelagenten. SPECTRE heeft  $N$  medewerkers, genummerd van 0 tot en met  $N - 1$ , en hun organigram is een boom met deze  $N$  punten.

MI6 zal een niet-lege verzameling  $S$  van SPECTRE's medewerkers kiezen om in dubbelagenten te veranderen. Vanwege het risico op verder verraad, is het essentieel dat de communicatielijnen tussen twee dubbelagenten alleen door 'onschuldige' medewerkers gaat. Oftewel, voor elk paar verschillende medewerkers  $a$  en  $b$  in  $S$ , bevat het simpele pad tussen  $a$  en  $b$  in de boom geen andere medewerkers in  $S$ .

Jouw opgave is om het aantal mogelijke verzamelingen  $S$  van dubbelagenten te tellen. Omdat deze waarde erg groot kan zijn, print de waarde modulo  $10^9 + 7$ .

## Implementatie

Je moet een enkel `.cpp` bestand inleveren.

 In de bijlagen van deze opgave vind je een template `trees.cpp` met een voorbeeld-implementatie.

Je moet de volgende functie implementeren:

```
C++ | int count_sets(int N, vector<int> U, vector<int> V);
```

- Integer  $N$  staat voor het aantal medewerkers van SPECTRE.
- De lijsten  $U$  en  $V$ , genummerd van 0 tot en met  $N - 2$ , bevatten de waarden  $U_0, U_1, \dots, U_{N-2}$  en  $V_0, V_1, \dots, V_{N-2}$ , waarbij  $U_i$  en  $V_i$  de uiteindes zijn van de  $i$ -de relatie van het organigram.
- De functie moet het aantal mogelijke verzamelingen modulo  $10^9 + 7$  teruggeven.

De grader zal de functie `count` aanroepen en de teruggegeven waarde printen naar het uitvoerbestand.

## Voorbeeld Grader

De map van de opgave bevat een versimpelde versie van de jury's grader, die je kan gebruiken om je oplossing lokaal te testen. De versimpelde grader leest de invoer van `stdin`, roept de functies aan die je moet implementeren en schrijft uiteindelijk de uitvoer naar `stdout`.

De invoer bestaat uit  $N$  regels, die bevatten:

- Regel 1: integer  $N$ .
- Regel  $2 + i$  ( $0 \leq i \leq N - 2$ ): de integers  $U_i$  en  $V_i$ .

De uitvoer bestaat uit een enkele regel, die de waarde teruggegeven door de functie `count` bevat.

## Randvoorwaarden

- $2 \leq N \leq 500\,000$ .
- $0 \leq U_i, V_i \leq N - 1$ .
- De knopen vormen een geldige boom.

## Scoring

Je programma wordt getest op een set testgevallen gegroepeerd per deelopgave. Om de score te krijgen die hoort bij een deelopgave, moet je alle testgevallen oplossen die daarbij horen.

- **Subtask 1** [ 0 punten]: Voorbeeld testgevallen.
- **Subtask 2** [20 punten]:  $N \leq 16$ .
- **Subtask 3** [15 punten]: De boom is een pad. Een pad is een boom waarbij de knopen herordend kunnen worden in een volgorde  $P_0, P_1, \dots, P_{N-1}$  waarbij er een verbinding is tussen knopen  $P_i$  en  $P_{i+1}$  voor alle  $0 \leq i \leq N - 2$ .
- **Subtask 4** [15 punten]:  $N \leq 500$ , en de boom is een rups. Een rups is een pad met wat extra bladeren eraan vast. Oftewel, de knopen van de boom kunnen geschreven worden als  $P_0, \dots, P_k, Q_0, \dots, Q_l$ , waarbij  $P_0, \dots, P_k$  een pad is en elke  $Q_i$  een blad (dus heeft graad 1).
- **Subtask 5** [15 punten]:  $N \leq 5000$ , en de boom is een rups.
- **Subtask 6** [10 punten]:  $N \leq 500\,000$ , en de boom is een rups.
- **Subtask 7** [25 punten]: Geen extra randvoorwaarden.

## Voorbeelden

stdin	stdout
3 0 1 1 2	6
5 0 1 0 2 1 3 1 4	17

## Uitleg

In het **eerste voorbeeld** is de boom een pad ( $0 - 1 - 2$ ). Er zijn 6 mogelijke verzamelingen in deze boom:  $\{0\}, \{1\}, \{0, 1\}, \{2\}, \{0, 2\}, \{1, 2\}$ .

De verzameling  $\{0, 1, 2\}$  is niet toegestaan omdat 1 op het pad tussen 0 en 2 ligt.

In het **tweede voorbeeld** is de boom een pad ( $3 - 1 - 0 - 2$ ) met één extra blad (4). Er zijn 17 mogelijke verzamelingen in deze boom.