Double Agents (`trees`)

L'agence anglaise d'espionage MI6 prévoit d'infiltrer un réseau d'agents doubles dans l'organisation criminelle SPECTRE. SPECTRE possède N employés numérotés de 0 à $N-1$ et son organigramme est un arbre de N noeuds.

MI6 sélectionnera un ensemble non vide S des employés SPECTRE pour en faire des agents doubles. Pour éviter tout risque de trahison (des possibles agents triples), il est essentiel que la chaîne de communication entre chaque paire d'agents doubles ne passent que par des employés 'innocents'. En d'autres termes, pour toute paire distincte d'employés a et b dans S, le chemin unique entre a et b dans l'arbre ne doit pas contenir d'autres employés dans S.

Votre tâche est de compter le nombre d'ensembles possibles S d'agents doubles. Vu que ce nombre peut être assez élevé, un modulo $10^9 + 7$ sera appliqué à la valeur en sortie

Implémentation

Vous devrez soumettre une seul fichier source `.cpp`.

> ☞ Parmi les fichiers fournis avec cette tâche, vous trouverez un fichier modèle `trees.cpp` avec une implémentation d'exemple.

Vous devez implémenter la fonction suivante :

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

- L'entier N correspond au nombre d'employés de SPECTRE.
- Les tableaux U et V, indexé de 0 to $N-2$, contiennent les valeurs $U_0, U_1, \ldots, U_{N-2}$ et $V_0, V_1, \ldots, V_{N-2}$, où U_i et V_i sont les extrémités de la i-ième arête dans l'organigramme.
- La fonction doit retourner le nombre d'ensembles possibles modulo $10^9 + 7$.

L'évaluateur (grader) appelera la fonction `count` et écrira son résultat dans le fichier de sortie.

Évaluateur

Le répertoire de la tâche contient une version simplifiée de l'évaluateur officiel, vous pouvez l'utiliser pour tester localement votre solution. L'évaluateur simplifié lit ses entrées de `stdin`, appelle la fonction que vous devez implémenter et écrit le résultat final dans `stdout`.

L'entée est composée de N lignes qui contiennent :

- Ligne 1 : l'entier N.
- Ligne $2 + i$ $(0 \leq i \leq N-2)$: les entiers U_i et V_i.

La sortie est composée d'une seule ligne, contenant la valeur retournée par la fonction `count`.

Contraintes

- $2 \leq N \leq 500\,000$.
- $0 \leq U_i, V_i \leq N-1$.

— Les noeuds forment un arbre valide.

Score

Votre programme sera testé sur un ensemble de tests groupés par sous-tâche. Pour obtenir le score associé à une sous-tâche, vous devez résoudre correctement l'ensemble des tests qu'elle contient.

— **Sous-tâche 1** [**0 points**]: Tests donnés en exemple.

— **Sous-tâche 2** [**20 points**]: $N \leq 16$.

— **Sous-tâche 3** [**15 points**]: L'arbre est une ligne. Une ligne est un arbre dans lequel tous les noeuds peuvent être arrangés dans un ordre $P_0, P_1, \ldots, P_{N-1}$ dans lequel il existe une arête entre les noeuds P_i et P_{i+1} pour tout $0 \leq i \leq N-2$.

— **Sous-tâche 4** [**15 points**]: $N \leq 500$, et l'arbre est une ligne avec quelques feuilles simples directement connectées à un noeud de la ligne.

— **Sous-tâche 5** [**15 points**]: $N \leq 5000$, et l'arbre est une ligne avec quelques feuilles simples directement connectées à un noeud de la ligne.

— **Sous-tâche 6** [**10 points**]: $N \leq 500\,000$, et l'arbre est une ligne avec quelques feuilles simples directement connectées à un noeud de la ligne.

— **Sous-tâche 7** [**25 points**]: Aucune contrainte supplémentaire.

Exemples

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

Explication

Dans le **premier exemple**, l'arbre est une ligne $(0-1-2)$. Il y a 6 ensembles possibles dans cet arbre : $\{0\}, \{1\}, \{0,1\}, \{2\}, \{0,2\}, \{1,2\}$.

L'ensemble $\{0,1,2\}$ n'est pas autorisé vu que 1 se trouve sur le chemin unique entre 0 et 2.

Dans le **second exemple**, l'arbre est une ligne $(3-1-0-2)$ avec une feuille supplémentaire (4). Il y a 17 ensembles possibles dans cet arbre.