

## Tâche 1.3 – Arbre à chats (cats) (100 pts)

Un immeuble proche du vôtre appartient à un gang de chats de gouttière musiciens, et vous êtes en charge du maintien du bâtiment.

Ce gang organise chaque soir à partir de minuit des sessions de jazz endiablées dans chacune des  $n$  pièces de l'immeuble, ce qui n'est pas sans conséquences sur l'état du bâtiment. En effet, le sol de chaque pièce peut résister seulement à un certain nombre  $r_i$  de chats, et si ce seuil est dépassé, le sol craque et tous les chats présents dans la pièce tombent dans la pièce  $p_i$  située juste en-dessous. Évidemment, si le sol de la pièce en-dessous est déjà cassé ou si son seuil de résistance est aussi dépassé, les chats continueront à tomber, et emporteront avec eux d'éventuels chats qui seraient présents sur leur chemin.

L'immeuble a une architecture particulière : en effet, il faut éviter à tout prix que des chats jouant dans une même pièce soient séparés lorsque le sol craque : en effet, ça les interromprait dans leurs improvisations musicales — un vrai désastre. Par conséquent chaque pièce (sauf le rez-de-chaussée) a exactement une pièce en-dessous d'elle, même si une pièce peut avoir plusieurs pièces au-dessus. De plus, en tombant suffisamment de fois à travers le sol, on atteint toujours le rez-de-chaussée, une pièce unique dont le sol peut résister à une infinité de chats (l'immeuble a donc une structure en arbre).

Pendant la soirée, vous observez l'arrivée de  $q$  groupes de chats un par un, par les fenêtres du bâtiment. Initialement, tous les sols de l'immeuble sont intacts (grâce à votre dur labeur) et il ne contient aucun chat. Une fois entrés dans une pièce, les chats ne bougeront jamais d'endroit, sauf évidemment si le sol de la pièce où ils se trouvent craque. Puisque l'approvisionnement en matériaux n'est pas une chose facile, vous aimeriez savoir au plus tôt combien de sols il faudra réparer pendant la journée pour remettre l'immeuble en état. Dès lors vous aimeriez savoir, après chaque entrée d'un groupe de chats, le nombre de sols actuellement cassés dans l'immeuble.

### Tâche

Écrivez un programme qui, étant donnée la structure du bâtiment, les résistances des sols et les arrivées des groupes de chats, détermine pour chaque groupe de chats le nombre total de sols cassés après leur arrivée (les sols déjà cassés auparavant sont aussi inclus).

### Limites et contraintes

- $1 \leq n \leq 10^5$ , le nombre de pièces ;
- $1 \leq q \leq 2 * 10^5$ , le nombre d'arrivées de groupes de chats observées ;
- $1 \leq r_i \leq 10^9$ , la résistance du sol de la pièce  $i$  ;
- $1 \leq s_j \leq 10^4$ , le nombre de chats dans le groupe  $j$ .

Pour les sous-tâches, on a les contraintes supplémentaires suivantes :

	Contraintes
Sous-tâche A (15 pts)	$n, q \leq 10^3$
Sous-tâche B (35 pts)	Toutes les pièces ont au plus une pièce au-dessus (les pièces forment une chaîne).
Sous-tâche C (50 pts)	Pas de contraintes supplémentaires.

Durée maximale d'exécution : **3 secondes**. Limite mémoire : **512 MB**.

**Attention** : Comme vous pouvez le voir, les sous-tâches sont assez diverses. Toutefois, votre score sera déterminé par votre soumission de score maximal, et non la somme des points de toutes les sous-tâches que vous avez résolues sur la durée du concours. Dès lors, il peut être nécessaire de tester les contraintes ci-dessus sur l'input et d'adapter le code que votre programme exécute en conséquence.

### Input

La première ligne de l'input contient deux nombres  $n, q$ , respectivement le nombre de pièces et le nombre de groupes de chats qui vont arriver. Les pièces sont numérotées de 0 à  $n - 1$ , où 0 est le numéro du rez-de-chaussée.

Les  $n - 1$  lignes suivantes contiennent chacune deux entiers :  $p_i$  et  $r_i$  ( $i \in \{1, \dots, n - 1\}$ ), où  $p_i$  est le numéro de

pièce en-dessous de  $i$  et  $r_i$  la résistance du sol de  $i$ .

Enfin, les  $q$  lignes suivantes contiennent chacune  $e_j$  et  $s_j$ , le numéro de la pièce d'entrée et le nombre de chat dans le  $j$ ème groupe.

### Output

Pour chacun des  $q$  groupes de chats, imprimez sur une ligne un seul entier : le nombre total de sols cassés dans le bâtiment après que le groupe  $y$  soit entré et que tous les chats qui devaient tomber soient tombés.

### Exemple 1

Input :

```
3 2
2 5
0 2
1 5
1 1
```

Output :

```
0
2
```

Dans cet exemple, il y a 3 pièces. La pièce 2 se trouve juste en-dessous la 1 et la 0 se trouve juste en-dessous de la 2. Le sol de la pièce 1 a une résistance de 5 et celui de la 2 a une résistance de 2. Deux groupes de chats entrent.

Le premier groupe entre dans la pièce 1 et comporte 5 chats. La résistance de la pièce 1 est égale à 5, donc le sol ne se casse pas (le sol ne se casserait que si strictement plus que 5 chats devaient se trouver dans la pièce). Après que le groupe de chats soit entré, il y a donc un total de 0 sols cassés dans le bâtiment.

Le groupe suivant entre aussi dans la pièce 1 et consiste d'un seul chat. Il y a donc maintenant au total 6 chats sur le sol de la pièce 1, et étant donné que sa résistance est 5, le sol cède sous le poids. Les chats tombent donc dans la pièce juste en-dessous de la pièce 1. Cette pièce est la 2, et son sol se casse également, car sa résistance est aussi strictement plus petite que 6. Les chats tombent donc ensuite dans la pièce 0 (le rez-de-chaussée), dont le sol ne se casse jamais (il peut supporter un nombre infini de chats). Au total, il y a donc maintenant 2 sols cassés dans le bâtiment.

### Exemple 2

Input :

```
1 3
0 10
0 100
0 1000
```

Output :

```
0
0
0
```

Dans cet exemple il y a une seule pièce, et 3 groupes de chats entrent. Les groupes entrent successivement, tous dans la pièce 0 le rez-de chaussée. Le sol de la pièce 0 peut supporter une infinité de chats et ne peut donc pas se casser. Par conséquent, après chaque arrivée, le nombre total de sols cassés dans le bâtiment est 0.

### Exemple 3

Input :

```
5 8
3 123
3 4
0 3
2 2
3 3
3 1
2 3
4 2
4 1
4 1
4 124
1 124
```

Output :

```
0
1
1
1
3
3
3
4
```