Stations (stations)

Le "Singapore Internet Backbone" (SIB) consiste en n stations, auxquelles sont assignés des **indices** de 0 à $n-1$. Il y a aussi $n-1$ câbles bidirectionnels, numérotés de 0 à $n-2$. Chaque câble connecte deux stations distinctes. Deux stations connectées via un câble direct sont dites voisines.

Un chemin de la station x à la station y est une suite de stations distinctes $a_0, a_1, \cdots, a_p$, telle que $a_0 = x$, $a_p = y$, et deux stations consécutives dans le chemin sont toujours voisines. Il y a **exactement un** chemin de toute station x à toute autre station y.

Toute station x peut créer un paquet (de données) et l'envoyer à une autre station y, qui est nommée la **cible** du paquet. Ce paquet doit transiter selon l'unique chemin de x à y de la manière suivante. Considérons une station z qui détient actuellement un paquet, dont la station cible est y ($z \neq y$). Dans cette situation, la station z :

1. éxecute une **procédure de routage** qui détermine le voisin de z qui est sur l'unique chemin de z à y, et
2. transfère le paquet à ce voisin.

Toutefois, les stations du SIB ont une mémoire limitée et ne stockent pas la liste entière des câbles du réseau pour l'utiliser lors de cette procédure.

Vous devez implémenter une procédure de routage pour le SIB, qui consiste en deux fonctions.

- La première fonction reçoit en entrée n, la liste des câbles du SIB et un entier $k \geq n-1$. Elle assigne à chaque station une **étiquette** unique, qui est un entier compris entre 0 et k, inclus.

- La deuxième fonction est la procédure de routage, et est déployée sur toutes les stations après que les étiquettes ont été assignées. Elle reçoit **uniquement** les entrées suivantes :

    - s, l'**étiquette** de la station qui détient actuellement un paquet,
    - t, l'**étiquette** de la station cible du paquet ($t \neq s$),
    - c, la liste des **étiquettes** de tous les voisins de s.

    Elle doit renvoyer l'**étiquette** du voisin de s à qui le paquet devrait être transféré.

Dans une des sous-tâches, le score de votre solution dépend de la valeur le l'étiquette maximale affectée à une station (plus elle est petite, plus vous aurez de points).

Détails d'implémentation

Vous devez implémenter les fonctions suivantes :

```
int[] label(int n, int k, int[] u, int[] v)
```

- n : nombre de stations dans le SIB.
- k : étiquette maximale autorisée.
- u et v : des tableaux de taille $n-1$ décrivant les câbles. Pour chaque i ($0 \leq i \leq n-2$), le câble i connecte les stations d'indices $u[i]$ et $v[i]$.
- Cette fonction doit renvoyer un tableau L de taille n. Pour chaque i ($0 \leq i \leq n-1$), $L[i]$ est l'étiquette assignée à la station d'indice i. Tous les éléments du tableau L doivent être uniques et compris entre 0 et k, inclus.

```
int find_next_station(int s, int t, int[] c)
```

- s : étiquette de la station détenant un paquet.
- t : étiquette de la station cible du paquet.
- c : un tableau donnant la liste des étiquettes de tous les voisins de s. Le tableau c est stocké par ordre croissant.
- Cette fonction doit renvoyer l'étiquette du voisin de s auquel le paquet doit être transféré.

Chaque test contient un ou plusieurs scénarios indépendants (c'est à dire des configurations différentes du SIB). Pour un test contenant r scénarios, un **programme** qui appelle ces deux fonctions est éxecuté deux fois, de la manière suivante.

Lors de la première exécution du programme :

- la fonction `label` est appelée r fois,
- les étiquettes renvoyées sont stockées par l'évaluateur, et
- `find_next_station` n'est pas appelée.

Lors de la seconde exécution du programme :

- `find_next_station` peut être appelée plusieurs fois. Pour chaque appel, un scénario **arbitraire** est choisi et les étiquettes renvoyées par l'appel à `label` dans ce scénario sont données en entrée à `find_next_station`
- `label` n'est pas appelée.

En particulier, aucune information stockée dans des variables statiques ou globales lors de la première exécution n'est accessible à la fonction `find_next_station`.

Exemple

Considérons l'appel suivant :

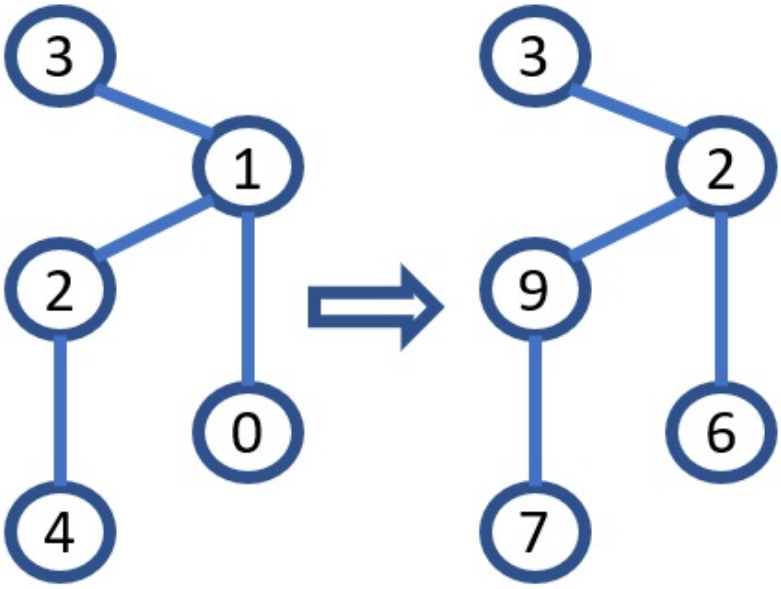
```
label(5, 10, [0, 1, 1, 2], [1, 2, 3, 4])
```

Il y a un total de 5 stations, et 4 câbles connectent les paires de stations avec les indices $(0, 1)$, $(1, 2)$, $(1, 3)$ et $(2, 4)$. Chaque étiquette peut être un entier entre 0 et $k = 10$.

Pour l'étiquetage suivant :

Indice	Étiquette
0	6
1	2
2	9
3	3
4	7

La fonction `label` doit renvoyer $[6, 2, 9, 3, 7]$. Les nombres dans la figure suivante montrent les indices (à gauche) et les étiquettes assignées (à droite).



Supposons que les étiquettes ont été assignées comme décrit ci-dessus et considérons l'appel suivant :

```
find_next_station(9, 6, [2, 7])
```

Cela signifie que la station détenant le paquet a l'étiquette 9, et la station cible a l'étiquette 6. Les étiquettes des stations sur le chemin vers la station cible sont $[9, 2, 6]$. Par conséquent, l'appel doit renvoyer 2, qui est l'étiquette de la station à laquelle le paquet doit être transférée (qui a l'indice 1).

Considérons un autre appel possible :

```
find_next_station(2, 3, [3, 6, 9])
```

La fonction doit renvoyer 3, car la station cible (qui a l'étiquette 3) est voisine de la station d'étiquette 2, et doit par conséquent recevoir le paquet directement.

Contraintes

- $1 \leq r \leq 10$

Pour chaque appel à `label` :

- $2 \leq n \leq 1000$
- $k \geq n - 1$
- $0 \leq u[i], v[i] \leq n - 1$ (pour tout $0 \leq i \leq n - 2$)

Pour chaque appel à `find_next_station`, l'entrée provient d'un appel précédent à `label` choisi arbitrairement. Considérons les étiquettes produites. On a :

- s et t sont les étiquettes de deux stations différentes.
- c est la suite de toutes les étiquettes des voisins de la station ayant l'étiquette s, par ordre croissant.

La longueur totale de tous les tableaux c passés en entrée à la procédure `find_next_station` n'excède pas $100\ 000$ pour tous les scénarios combinés.

Sous-tâches

1. (5 points) $k = 1000$, aucune station n'a plus de 2 voisins.
2. (8 points) $k = 1000$, le câble i connecte les stations $i + 1$ et $\left\lfloor \frac{i}{2} \right\rfloor$.
3. (16 points) $k = 1\ 000\ 000$, au plus une station a plus de deux voisins.
4. (10 points) $n \leq 8$, $k = 10^9$
5. (61 points) $k = 10^9$

Dans la sous-tâches 5, vous pouvez obtenir un score partiel. Soit m la valeur de l'étiquette maximale renvoyée par `label` parmi tous les scénarios. Votre score pour cette sous-tâche est calculé selon le tableau suivant :

Étiquette maximale	Score
$m \geq 10^9$	0
$2000 \leq m < 10^9$	$50 \cdot \log_{5 \cdot 10^5}(\frac{10^9}{m})$
$1000 < m < 2000$	50
$m \leq 1000$	61

Évaluateur d'exemple

L'évaluateur d'exemple lit l'entrée au format suivant :

- ligne 1: r

Suivent r blocs, chacun décrivant un scénario. Le format de chaque bloc est le suivant :

- ligne 1 : n k
- ligne $2 + i$ $(0 \leq i \leq n - 2)$: $u[i]$ $v[i]$
- ligne $1 + n$: q, le nombre d'appels à `find_next_station`.
- ligne $2 + n + j$ $(0 \leq j \leq q - 1)$: $z[j]$ $y[j]$ $w[j]$: les **indices** des stations impliquées dans le j-ème appel à `find_next_station`. La station $z[j]$ détient le paquet, la station $y[j]$ est la cible du paquet, et la station $w[j]$ est la station à laquelle le paquet doit être transféré.

L'évaluateur d'exemple affiche le résultat au format suivant :

- ligne 1 : m

Suivent r blocs correspondant aux scénarios de l'entrée. Le format de chaque bloc est le suivant :

- ligne $1 + j$ $(0 \leq j \leq q - 1)$: **indice** de la station dont l'**étiquette** a été renvoyée par le j-ème appel à `find_next_station` dans ce scénario.

Notez que chaque exécution de l'évaluateur d'exemple appelle `label` et `find_next_station`.