Heure de fermeture

La Hongrie est un pays avec N villes, numérotées de 0 à $N-1$.

Les villes sont reliées par $N-1$ routes *bidirectionnelles*, numérotées de 0 à $N-2$. Pour chaque j tel que $0 \le j \le N-2$, la route j relie la ville $U[j]$ à la ville $V[j]$ et a pour longueur $W[j]$, autrement dit, elle permet de voyager entre les villes en $W[j]$ unités de temps. Chaque route relie deux villes différentes, et chaque paire de villes est reliée par au plus une route.

Un **chemin** entre deux villes distinctes a et b est une suite $p_0, p_1, \ldots, p_t$ de villes distinctes, telle que :

- $p_0 = a$,
- $p_t = b$,
- pour chaque i ($0 \le i < t$), il y a une route qui relie les villes p_i et p_{i+1}.

Il est possible de voyager de n'importe quelle ville à n'importe quelle autre en utilisant des routes, autrement dit, il existe un chemin entre chaque paire de villes distinctes. On peut montrer que ce chemin est unique pour chaque paire de villes distinctes.

La **longueur** d'un chemin $p_0, p_1, \ldots, p_t$ est la somme des longueurs des t routes qui relient les villes consécutives le long du chemin.

En Hongrie, de nombreuses personnes voyagent pour assister aux festivités de la Fête de Fondation de l'État dans deux grandes villes. Une fois que les célébrations sont terminées, elles retournent chez elles. Le gouvernement souhaite éviter que la foule dérange les habitants locaux, ils planifient donc d'instaurer un couvre-feu dans toutes les villes à une certaine heure. À chaque ville sera attribué un entier positif ou nul par le gouvernement, **l'heure de fermeture**. Le gouvernement a décidé que la somme de toutes les heures de fermeture ne doit pas dépasser K. Plus précisément, pour chaque i entre 0 et $N-1$ inclus, l'heure de fermeture attribuée à la ville i est un entier positif ou nul $c[i]$. La somme des $c[i]$ doit être inférieure ou égale à K.

Considérons une ville a, ainsi qu'une attribution des heures de fermeture. On dit que la ville b est **atteignable** depuis la ville a si et seulement si $b = a$, ou le chemin $p_0, \ldots, p_t$ entre ces deux villes (donc en particulier $p_0 = a$ et $p_t = b$) vérifie les conditions suivantes :

- la longueur du chemin p_0, p_1 est d'au plus $c[p_1]$, et
- la longueur du chemin p_0, p_1, p_2 est d'au plus $c[p_2]$, et
- $\ldots$

- la longueur du chemin $p_0, p_1, p_2, \ldots, p_t$ est d'au plus $c[p_t]$.

Cette année, les deux lieux principaux de festivité sont situés dans la ville X et la ville Y. Pour chaque attribution des heures de fermeture, le **score de praticité** est défini comme la somme des deux nombres suivants :

- Le nombre de villes atteignables depuis la ville X.
- Le nombre de villes atteignables depuis la ville Y.

Notez que si une ville est à la fois atteignable depuis la ville X et depuis la ville Y, elle compte *deux fois* dans le calcul du score de praticité.

Votre tâche est de calculer le score de praticité maximum qu'il est possible d'atteindre avec une attribution bien choisie des heures de fermetures.

Détails d'implémentation

Vous devez implémenter la fonction suivante.

```
int max_score(int N, int X, int Y, int64 K, int[] U, int[] V, int[] W)
```

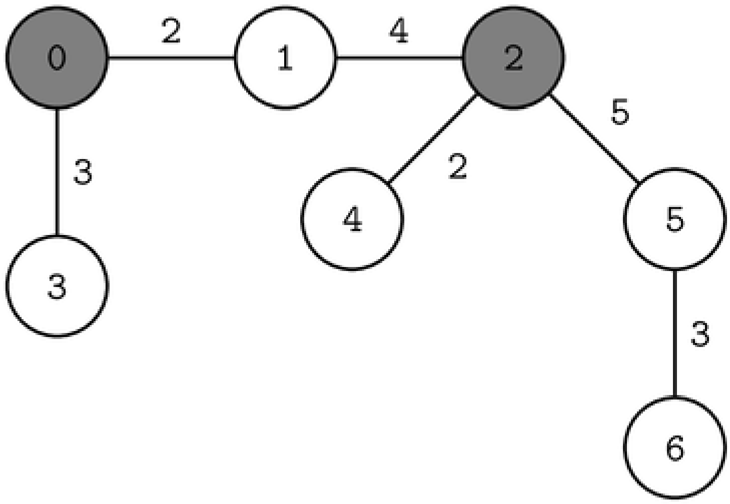
- N: le nombre de villes.
- X, Y: les villes où sont situés les lieux principaux de festivité.
- K: la borne supérieure sur la somme des heures de fermeture.
- U, V: deux tableaux de longueur $N - 1$ qui décrivent les villes que relient les routes.
- W: un tableau de longueur $N - 1$ qui décrit les longueurs des routes.
- Cette fonction doit renvoyer le score de praticité maximum qui peut être atteint par une attribution valide des heures de fermeture.
- Cette fonction peut être appelée **plusieurs fois** pour chaque test.

Exemple

Considérons l'appel suivant :

```
max_score(7, 0, 2, 10,
          [0, 0, 1, 2, 2, 5], [1, 3, 2, 4, 5, 6], [2, 3, 4, 2, 5, 3])
```

Cela correspond au réseau de routes suivant :



Supposons que les heures de fermeture sont attribuées comme suit :

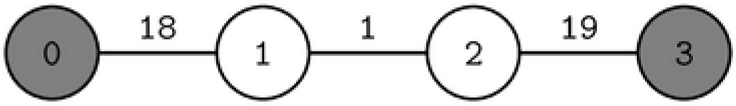
Ville	0	1	2	3	4	5	6
Heure de fermeture	0	4	0	3	2	0	0

Notez que la somme des heures de fermeture est 9, qui est inférieur ou égal à $K = 10$. Les villes $0, 1$, et 3 sont atteignables depuis la ville X ($X = 0$), tandis que les villes 1, 2, et 4 sont atteignables depuis la ville Y ($Y = 2$). Ainsi, le score de praticité est de $3 + 3 = 6$. Il n'existe pas d'attribution des heures de fermeture qui donne un score de praticité supérieur à 6. La fonction doit donc renvoyer 6.

Considérons également l'appel suivant :

```
max_score(4, 0, 3, 20, [0, 1, 2], [1, 2, 3], [18, 1, 19])
```

Cela correspond au réseau de routes suivant :



Supposons que les heures de fermeture sont attribuées comme suit :

Ville	0	1	2	3
Heure de fermeture	0	1	19	0

La ville 0 est atteignable depuis la ville X ($X = 0$), tandis que les villes 2 et 3 sont atteignables depuis la ville Y ($Y = 3$). Ainsi, le score de praticité est de $1 + 2 = 3$. Il n'existe pas d'attribution des heures de fermeture qui donne un score de praticité supérieur à 3. La fonction doit donc renvoyer 3.

Contraintes

- $2 \le N \le 200\,000$
- $0 \le X < Y < N$
- $0 \le K \le 10^{18}$
- $0 \le U[j] < V[j] < N$ (pour chaque j tel que $0 \le j \le N - 2$)
- $1 \le W[j] \le 10^6$ (pour chaque j tel que $0 \le j \le N - 2$)
- Il est possible de voyager de chaque ville à chaque autre en utilisant les routes.
- $S_N \le 200\,000$, où S_N est la somme des N pour tous les appels à `max_score` dans chaque test.

Sous-tâches

On dit qu'un réseau de routes est **linéaire** si la route i relie les villes i et $i + 1$ (pour chaque i tel que $0 \le i \le N - 2$).

1. (8 points) La longueur du chemin entre la ville X et la ville Y est strictement plus grande que $2K$.
2. (9 points) $S_N \le 50$, le réseau de routes est linéaire.
3. (12 points) $S_N \le 500$, le réseau de routes est linéaire.
4. (14 points) $S_N \le 3\,000$, le réseau de routes est linéaire.
5. (9 points) $S_N \le 20$
6. (11 points) $S_N \le 100$
7. (10 points) $S_N \le 500$
8. (10 points) $S_N \le 3\,000$
9. (17 points) Aucune contrainte supplémentaire.

Évaluateur d'exemple (grader)

Notons C le nombre de scénarios, c'est à dire, le nombre d'appels à `max_score`. L'évaluateur lit l'entrée dans le format suivant :

- ligne 1: C

Les descriptions des C scénarios suivent.

L'évaluateur lit la description de chaque scénario dans le format suivant :

- ligne 1: $N\ X\ Y\ K$

- ligne $2 + j$ ($0 \leq j \leq N - 2$): $U[j]$ $V[j]$ $W[j]$

L'évaluateur affiche une unique ligne pour chaque scénario, dans le format suivant :

- ligne 1: la valeur de retour de `max_score`