



Problème Récolte de Diamants

 3 sec.  4 Mo

Une mine de diamants a été découverte dans les montagnes des Rhodopes. Par souci de simplicité, nous supposons que la mine comporte N salles, étiquetées par les entiers de 0 à $N - 1$. Il y a M couloirs à sens unique reliant certaines salles, de sorte qu'au moins un couloir parte de chaque salle. Chaque couloir a un nombre de diamants que l'on récolte en le traversant. Ce nombre *ne change pas* lorsque l'on emprunte le couloir : il reste le même pour les passages suivants.

Il est possible qu'un couloir relie une salle à elle-même, et qu'il existe plusieurs couloirs entre deux salles (éventuellement dans la même direction). Il n'est pas non plus garanti que les salles soient connectées ; il peut donc exister une paire de salles (x, y) telle que y ne puisse être atteinte depuis x .

Petar traversera K couloirs pour récolter des diamants. Il choisira une salle s pour commencer, puis se déplacera vers une autre salle en empruntant un couloir partant de s , et ainsi de suite jusqu'à avoir parcouru exactement K couloirs. Notez qu'il peut repasser par les mêmes couloirs et salles, et que le nombre de diamants récoltés dans un couloir ne change pas lors des passages. Observez qu'il existe toujours un moyen de traverser K couloirs consécutivement.

Petar choisira s et le chemin qu'il suivra de la manière suivante. Il souhaite d'abord maximiser le nombre de diamants qu'il récoltera dans le premier couloir qu'il emprunte. Parmi toutes ces options, il maximisera le nombre de diamants qu'il récoltera dans le second couloir. Ceci répété K fois. Autrement dit, Petar souhaite choisir un chemin lexicographiquement maximal. Il se demande quel sera le nombre total de diamants qu'il récoltera s'il choisit un tel chemin. Aidez-le à le calculer.



Détails d'implémentation

Vous devez implémenter la fonction `calculate_diamonds` :

```
long long int calculate_diamonds(int N, int M, int K,  
    std::vector<int> u, std::vector<int> v, std::vector<int> d)
```

- N : le nombre de salles dans la mine de diamants ;
- M : le nombre de couloirs entre les salles ;
- K : le nombre de couloirs que Petar parcourra ;
- u, v, d : vectors de M entiers, représentant les salles d'entrée, salles de sortie, et nombres de diamants des couloirs.

Cette fonction sera appelée une fois par test et doit renvoyer un nombre, le nombre total de diamants récoltés en utilisant sa stratégie.



Contraintes

- $1 \leq N \leq 2\,000$
- $1 \leq M \leq 4\,000$
- $1 \leq K \leq 10^9$
- $0 \leq u[i], v[i] < N$
- $1 \leq d[i] \leq 10^9$ pour chaque $0 \leq i < M$
- Il est garanti qu'il existe au moins un couloir partant de chaque salle.
- **Notez que la limite mémoire de 4 Mo est inhabituellement petite.**



Sous-tâches

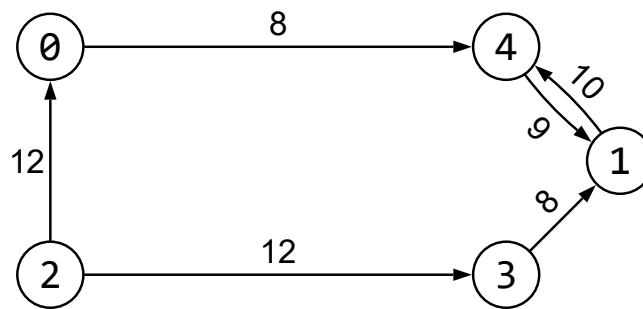
Sous-tâche	Points	Sous-tâches requises	N	M	K	Contraintes supplémentaires
0	0	—	—	—	—	Les exemples.
1	11	0	≤ 10	≤ 20	≤ 10	—
2	10	0 – 1	≤ 100	$\leq 1\,000$	≤ 1000	—
3	26	0 – 2	≤ 100	$\leq 1\,000$	$\leq 10^9$	—
4	11	—	$\leq 2\,000$	$= N$	$\leq 10^9$	Chaque salle a exactement un couloir sortant et un couloir entrant.
5	10	—	$\leq 2\,000$	$\leq 4\,000$	$\leq 10^9$	Tous les $d[i]$ sont distincts.
6	11	—	$\leq 2\,000$	$\leq 4\,000$	$\leq 10^9$	Il y a exactement un $d[i] = 2$ ($0 \leq i < M$) et toutes les autres valeurs de d sont égales à 1.
7	21	0 – 6	$\leq 2\,000$	$\leq 4\,000$	$\leq 10^9$	—



Exemple 1

Considérez l'appel suivant, illustré ci-dessous, avec $N = 5$, $M = 6$ et $K = 4$:

```
calculate_diamonds(5, 6, 4,
    {2, 0, 4, 2, 3, 1}, {0, 4, 1, 3, 1, 4}, {12, 8, 9, 12, 8, 10})
```



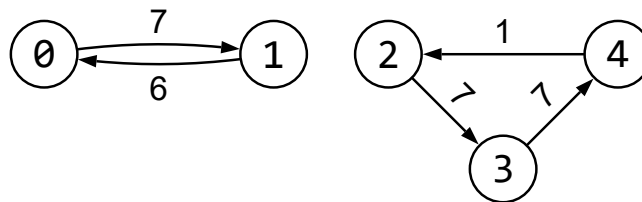
Petar choisira de passer par les couloirs suivants : $2 \xrightarrow{12} 3 \xrightarrow{8} 1 \xrightarrow{10} 4 \xrightarrow{9} 1$. Le nombre total de diamants qu'il récoltera est 39, qui doit être la valeur renvoyée par l'appel de fonction.



Exemple 2

Considérez l'appel suivant, illustré ci-dessous, avec $N = 5$, $M = 5$ et $K = 4$:

```
calculate_diamonds(5, 5, 4,
    {0, 1, 2, 3, 4}, {1, 0, 3, 4, 2}, {7, 6, 7, 7, 1})
```



Il y a 5 possibilités de chemins passant par 4 couloirs :

- (1) $0 \xrightarrow{7} 1 \xrightarrow{6} 0 \xrightarrow{7} 1 \xrightarrow{6} 0$;
- (2) $1 \xrightarrow{6} 0 \xrightarrow{7} 1 \xrightarrow{6} 0 \xrightarrow{7} 1$;
- (3) $2 \xrightarrow{7} 3 \xrightarrow{7} 4 \xrightarrow{1} 2 \xrightarrow{7} 3$;
- (4) $3 \xrightarrow{7} 4 \xrightarrow{1} 2 \xrightarrow{7} 3 \xrightarrow{7} 4$;
- (5) $4 \xrightarrow{1} 2 \xrightarrow{7} 3 \xrightarrow{7} 4 \xrightarrow{1} 2$.

Les chemins (2) et (5) ne maximisent pas le nombre de diamants dans le premier couloir. Parmi les chemins (1), (3) et (4), seul le chemin (3) maximise le nombre de diamants dans le second couloir, il s'agit donc du meilleur chemin selon Petar. Notez que le chemin (3) ne maximise ni le nombre de diamants dans le troisième couloir, ni le nombre total de diamants récoltés, mais il s'agit de l'unique chemin lexicographiquement maximal. Le nombre total de diamants récoltés par Petar sera 22, qui doit être la valeur renvoyée par l'appel de fonction.



Évaluateur d'exemple

Le format d'entrée est le suivant :

- ligne 1 : trois entiers, les valeurs de N , M et K .



- ligne $1 + i$: trois entiers $u[i]$, $v[i]$ et $d[i]$, représentant un couloir de la salle $u[i]$ à la salle $v[i]$ avec $d[i]$ diamants.

Le format de sortie est le suivant :

- ligne 1 : un entier, la valeur renvoyée par l'appel de fonction.