

C. Sopsug

Nom du problème	Sopsug
Limite de temps	5 secondes
Limite de mémoire	1 gigaoctet

Grushög est un quartier résidentiel en construction dans la banlieue de Lund. A présent, toutes les infrastructures nécessaires ont déjà été construites, ce qui comprend une infrastructure de la plus haute importance : celle de collecte des déchets. Comme c'est souvent le cas en Suède, un *sopsug* (système de collecte pneumatique des déchets) est utilisé pour ramasser les ordures. L'idée est de transporter les ordures dans des tubes enfouis sous terre, en utilisant de l'air comprimé.

Il y a N bâtiments dans la ville de Grushög, numérotés de 0 à $N - 1$. Vous pouvez connecter certaines paires de bâtiments avec des tubes. Si un tube relie le bâtiment u au bâtiment v , alors tous les déchets de u seront envoyés à v (mais pas dans l'autre direction). Votre but est de créer un réseau de $N - 1$ tubes tel que toutes les ordures finissent dans un même bâtiment. Autrement dit, vous voulez que votre réseau forme un arbre enraciné, avec toutes les arêtes orientées vers la racine.

Cependant, M tubes ont déjà été construits entre les bâtiments. Ces tubes *doivent* être utilisés dans votre réseau. Ces tubes sont orientés, et ne peuvent donc être utilisés que dans une unique direction déterminée.

De plus, il existe K paires de bâtiments entre lesquels il est impossible de construire un tube. Ces paires sont ordonnées, par conséquent même s'il est impossible de construire un tube de u à v , il pourrait être possible de construire un tube de v à u .

Entrée

La première ligne de l'entrée contient trois entiers, N , M , et K .

Les M lignes suivantes contiennent deux entiers distincts a_i, b_i , ce qui signifie qu'il existe déjà un tube de a_i à b_i .

Les K lignes suivantes contiennent deux entiers distincts c_i, d_i , ce qui signifie qu'il est impossible de construire un tube de c_i à d_i .

Toutes les $M + K$ paires ordonnées de l'entrée sont distinctes. Notez que (u, v) et (v, u) sont considérées comme des paires différentes.

Sortie

S'il n'existe pas de solution, affichez "NO".

Sinon, affichez $N - 1$ lignes, chacune contenant deux entiers u_i, v_i , ce qui signifie qu'il devrait y avoir un tube orienté de u_i à v_i . Vous pouvez afficher les tubes dans n'importe quel ordre. S'il existe plusieurs solutions, vous pouvez afficher n'importe laquelle. N'oubliez pas que les M tubes déjà construits doivent être inclus dans votre solution.

Contraintes et Score

- $2 \leq N \leq 300\,000$.
- $0 \leq M \leq 300\,000$.
- $0 \leq K \leq 300\,000$.
- $0 \leq a_i, b_i \leq N - 1$ pour $i = 0, 1, \dots, M - 1$.
- $0 \leq c_i, d_i \leq N - 1$ pour $i = 0, 1, \dots, K - 1$.

Votre solution sera testée sur un ensemble de groupes de tests (sous-tâches), chacun valant un certain nombre de points. Chaque sous-tâche contient un ensemble de tests. Pour obtenir tous les points d'une sous-tâche, vous devez valider tous les tests de cette sous-tâche.

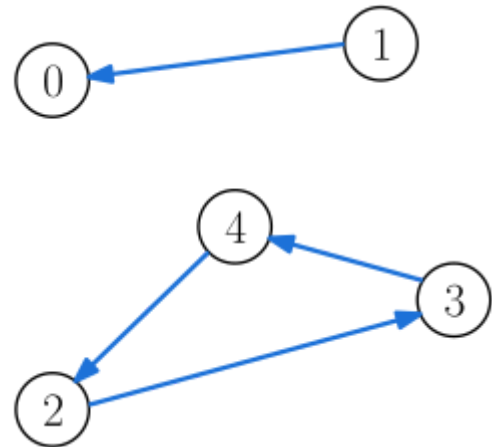
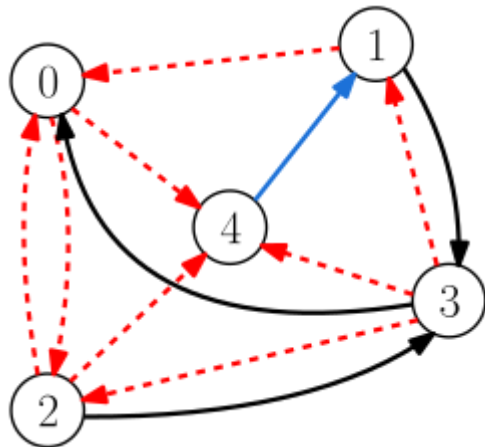
Sous-tâche	Score	Contraintes
1	12	$M = 0$ et $K = 1$
2	10	$M = 0$ et $K = 2$
3	19	$K = 0$
4	13	$N \leq 100$
5	17	Il est garanti qu'il existe une solution ayant 0 pour racine
6	11	$M = 0$
7	18	Pas de contraintes additionnelles

Exemple

Les figures ci-dessous illustrent le premier et deuxième exemple de test. Les arêtes bleues représentent les tubes déjà construits, et les arêtes en rouge pointillé représentent les tubes qu'il est impossible de construire.

La figure à gauche correspond à la solution donnée dans l'exemple de test, les arêtes noires étant les tubes (qui sont ajoutés en plus du tube déjà construit de 4 à 1, représenté en bleu). Dans ce réseau, toutes les ordures sont collectées dans le bâtiment 0. Cela n'est pas la seule solution ; par exemple, le tube de 1 à 3 peut être remplacé par un tube de 0 à 1, ce qui est aussi une solution valide.

Pour le deuxième exemple de test, on peut voir dans la figure de droite qu'il est impossible de construire une solution en raison du cycle (2, 3, 4).



Input	Output
<pre> 5 1 8 4 1 3 1 3 4 3 2 0 2 0 4 2 4 1 0 2 0 </pre>	<pre> 4 1 3 0 1 3 2 3 </pre>
<pre> 5 4 0 1 0 2 3 3 4 4 2 </pre>	<p>NO</p>
<pre> 3 0 1 0 1 </pre>	<pre> 1 0 2 0 </pre>
<pre> 4 0 2 0 1 1 0 </pre>	<pre> 2 0 3 0 1 3 </pre>