

Expo de Jeux de Société

Chaque année, une grande Expo de Jeux de Société a lieu à Cluj-Napoca, présentant une large sélection de nouveaux jeux. L'attraction principale cette année est un jeu appelé BoardOina.

Il y a n joueurs alignés dans une file, attendant d'essayer le jeu. Les joueurs sont numérotés de 0 à $n - 1$ selon leur position dans la file. Le joueur 0 est en tête de file et le joueur $n - 1$ est en fin de file.

Il existe m **relations d'amitié distinctes** entre m paires de joueurs dans la file. Plus précisément, pour chaque i de 0 à $m - 1$ inclus, le joueur $x[i]$ et le joueur $y[i]$ sont amis, où $0 \leq x[i] < y[i] < n$. Les relations d'amitié sont symétriques.

Considérons une séquence de k joueurs *consécutifs* dans la file commençant au joueur s (pour tout s et k tels que $0 \leq s < n$ et $1 \leq k \leq n - s$). Cette séquence de joueurs forme un **groupe d'amis** de taille k si, pour toute paire de deux joueurs, ils sont connectés par une séquence de relations d'amitié à l'intérieur de ce groupe d'amis. Plus précisément, les joueurs $s, s + 1, \dots, s + k - 1$ forment un groupe d'amis de taille k si, pour chaque u et v tels que $s \leq u < v < s + k$, il existe une séquence de joueurs $p[0], \dots, p[l - 1]$ telle que :

- $l \geq 2$;
- $s \leq p[j] < s + k$ pour chaque j de 0 à $l - 1$ inclus ;
- $p[0] = u$ et $p[l - 1] = v$;
- les joueurs $p[j]$ et $p[j + 1]$ sont amis pour chaque j de 0 à $l - 2$ inclus.

Notez que dans le cas où $k = 1$, le joueur s seul forme un groupe d'amis de taille 1.

BoardOina peut être joué par n'importe quel nombre de joueurs. Cependant, pour assurer le succès du jeu, les organisateurs autorisent uniquement les groupes d'amis à y jouer.

Un seul groupe peut jouer à la fois. Pour chaque partie, un groupe d'amis est formé à partir du joueur en tête de file, et commence à jouer. Les joueurs de ce groupe d'amis sont ensuite retirés de la file. Ce processus est répété jusqu'à ce que la file soit vide. Formellement, on dit que la file **peut être partitionnée en g groupes d'amis** s'il existe un tableau de tailles de groupes, $K = [K[0], K[1], \dots, K[g - 1]]$, tel que chacune des conditions suivantes soit remplie :

- $g > 0$ et $K[j] > 0$ (pour chaque j tel que $0 \leq j < g$) ;
- $K[0] + K[1] + \dots + K[g - 1] = n$;

- pour chaque j entre 0 et $g - 1$ inclus, les joueurs $s[j], s[j] + 1, \dots, s[j] + K[j] - 1$ forment un groupe d'amis de taille $K[j]$, où $s[0] = 0$ et sinon $s[j] = K[0] + K[1] + \dots + K[j - 1]$.

Les organisateurs veulent *minimiser* le nombre de groupes d'amis qui jouent au jeu. C'est-à-dire qu'ils souhaitent partitionner la file en g groupes d'amis de sorte qu'il ne soit pas possible de la partitionner en $g - 1$ (ou moins) groupes d'amis.

Votre tâche consiste à trouver une partition de la file en un nombre minimal de groupes d'amis, et à renvoyer le tableau des tailles des groupes.

Détails d'implémentation

Vous devez implémenter la procédure suivante.

```
std::vector<int> partition_players(int n, int m, std::vector<int> X,
                                std::vector<int> Y)
```

- n : le nombre de joueurs dans la file.
- m : le nombre de relations d'amitié.
- x, y : tableaux de longueur m décrivant les relations d'amitié.
- Cette procédure doit renvoyer un tableau des tailles des groupes, représentant une partition de la file de joueurs en un nombre minimal de groupes d'amis.
- Cette procédure est appelée exactement une fois pour chaque cas de test.

Limites

- $2 \leq n \leq 100\,000$
- $0 \leq m \leq 200\,000$
- $0 \leq x[i] < y[i] < n$ (pour chaque i tel que $0 \leq i < m$)
- Les relations d'amitié sont distinctes. Autrement dit, $x[i] \neq x[j]$ ou $y[i] \neq y[j]$ (pour chaque i et j tels que $0 \leq i < j < m$).
- S'il existe plusieurs solutions avec un nombre minimal de groupes, vous pouvez renvoyer n'importe quelle solution valide.

Sous-tâches

1. (5 points) $y[i] = x[i] + 1$ pour chaque i de 0 à $m - 1$ inclus.
2. (7 points) $y[i] \leq x[i] + 2$ pour chaque i de 0 à $m - 1$ inclus.
3. (6 points) $n \leq 300$ et $m \leq 600$
4. (15 points) $n \leq 2\,000$ et $m \leq 4\,000$
5. (34 points) Il n'existe aucune relation d'amitié *cyclique*. C'est-à-dire que, pour toute séquence de joueurs *distincts* $p[0], \dots, p[l - 1]$ telle que $l \geq 3$ et que, pour chaque $0 \leq j < l - 1$, les joueurs $p[j]$ et $p[j + 1]$ soient amis, les joueurs $p[0]$ et $p[l - 1]$ ne sont **pas** amis.
6. (33 points) Aucune contrainte supplémentaire.

Exemples

Exemple 1

Considérons l'appel suivant :

```
partition_players(5, 3, {0, 1, 3}, {1, 4, 4})
```

Dans cet exemple, les joueurs 0 et 1, les joueurs 1 et 4, ainsi que les joueurs 3 et 4 sont amis.

Le joueur 2 n'a aucun ami dans la file, il doit donc former à lui seul un groupe d'amis, ce qui implique que le nombre minimal de groupes d'amis est $g = 3$. Par ailleurs, les joueurs 0 et 1, ainsi que les joueurs 3 et 4, peuvent former chacun un groupe d'amis de taille 2.

Par conséquent, la file peut être partitionnée en 3 groupes d'amis de tailles 2, 1 et 2, donc la procédure peut renvoyer $[2, 1, 2]$.

Exemple 2

Considérons l'appel suivant :

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

Dans cet exemple, les joueurs 0 et 1, les joueurs 4 et 5, les joueurs 2 et 4, les joueurs 1 et 5, les joueurs 2 et 5, ainsi que les joueurs 3 et 6 sont amis.

Le seul ami du joueur 3 est le joueur 6, donc tout groupe d'amis contenant le joueur 3 est soit :

- un groupe d'amis de taille 1 contenant le joueur 3 seul, soit
- un groupe d'amis contenant à la fois le joueur 3 et le joueur 6.

Un groupe d'amis dans le second cas devrait également contenir les joueurs 4 et 5. Cela n'est pas possible puisque le seul ami du joueur 6 est le joueur 3, donc le joueur 3 n'est pas connecté aux joueurs 4 et 5 par une séquence de relations d'amitié.

Par conséquent, le joueur 3 doit être placé dans un groupe d'amis de taille 1. De même, le joueur 6 doit aussi être placé dans un groupe d'amis de taille 1, donc le nombre de groupes d'amis dans une partition est au moins 4.

Les joueurs 0, 1 et 2 ne forment pas un groupe d'amis de taille 3, car ni le joueur 0 ni le joueur 1 n'est connecté au joueur 2 par une séquence de relations d'amitié à l'intérieur du groupe. Ce problème ne se poserait pas si le joueur 5 était inclus dans le groupe, mais puisque les joueurs 3 et 4 seront nécessairement dans des groupes différents, cela n'arrivera jamais. Ainsi, le nombre de groupes d'amis dans une partition est au moins 5.

Par ailleurs, les joueurs 0 et 1, ainsi que les joueurs 4 et 5, forment deux groupes d'amis de taille 2. La file peut donc être partitionnée en 5 groupes d'amis de tailles 2, 1, 1, 2 et 1. La procédure peut ainsi renvoyer $[2, 1, 1, 2, 1]$.

Grader d'exemple

Le grader d'exemple lit l'entrée dans le format suivant :

- ligne 1 : $n \ m$
- ligne $2 + i$ ($0 \leq i < m$) : $x[i] \ y[i]$

Soient les éléments du tableau retourné par `partition_players` notés $K[0], K[1], \dots, K[g-1]$ pour un certain $g \geq 0$. La sortie du grader d'exemple est au format suivant :

- ligne 1 : g
- ligne 2 : $K[0] \ K[1] \ \dots \ K[g-1]$