Plus longue excursion

Les organisateurs des IOI 2023 ont de gros ennuis ! Ils ont oublié de planifier l'excursion à Ópusztaszer pour la journée à venir. Mais il n'est peut-être pas encore trop tard...

Il y a N monuments à Ópusztaszer numérotés de 0 à $N-1$. Certaines paires de ces monuments sont reliées par des **routes** *bidirectionnelles*. Chaque paire de monuments est reliée par au plus une route. Les organisateurs *ne savent pas* quels monuments sont reliés par des routes.

On dit que la **densité** du réseau routier à Ópusztaszer est **d'au moins** δ si tout ensemble de 3 monuments distincts a au moins δ routes les reliant. Autrement dit, pour tout triplet de monuments (u, v, w) tel que $0 \leq u < v < w < N$, parmi les paires de monuments $(u, v), (v, w)$ et (u, w), au moins δ paires sont reliées par une route.

Les organisateurs *connaissent* un entier strictement positif D tel que la densité du réseau routier est d'au moins D. Notez que la valeur de D ne peut pas être strictement plus grande que 3.

Les organisateurs peuvent faire des **appels** à la centrale téléphonique d'Ópusztaszer pour recueillir des informations sur les routes entre certains monuments. À chaque appel, deux tableaux non-vides de monuments $[A[0], \ldots, A[P-1]]$ et $[B[0], \ldots, B[R-1]]$ doivent être spécifiés. Les monuments doivent être deux-à-deux distincts, c'est à dire,

- $A[i] \neq A[j]$ pour tous i et j tels que $0 \leq i < j < P$;
- $B[i] \neq B[j]$ pour tous i et j tels que $0 \leq i < j < R$;
- $A[i] \neq B[j]$ pour tous i et j tels que $0 \leq i < P$ et $0 \leq j < R$.

Pour chaque appel, la centrale indique s'il y a une route reliant un monument de A et un monument de B. Plus précisément, la centrale itère sur toutes les paires i et j telles que $0 \leq i < P$ et $0 \leq j < R$. Si, pour une d'entre elles, les monuments $A[i]$ et $B[j]$ sont reliés par une route, la centrale renvoie `true`. Sinon, la centrale renvoie `false`.

Une **excursion** de longueur l est une séquence de monuments *distincts* $t[0], t[1], \ldots, t[l-1]$ pour laquelle, pour tout i entre 0 et $l-2$ inclus, le monument $t[i]$ et le monument $t[i+1]$ sont reliés par une route. Une excursion de longueur l est appelée une **plus longue excursion** s'il n'existe aucune excursion de longueur au moins $l+1$.

Votre tâche est d'aider les organisateurs à trouver une plus longue excursion à Ópusztaszer en faisant des appels à la centrale.

Détails d'implémentation

Vous devez implémenter la fonction suivante :

```
int[] longest_trip(int N, int D)
```

- N : le nombre de monuments à Ópusztaszer.
- D : la densité minimale garantie du réseau routier.
- Cette fonction doit renvoyer un tableau $t = [t[0], t[1], \ldots, t[l-1]]$, représentant une plus longue excursion.
- Cette fonction peut être appelée **plusieurs fois** dans chaque test.

La fonction précédente peut faire des appels à la fonction suivante :

```
bool are_connected(int[] A, int[] B)
```

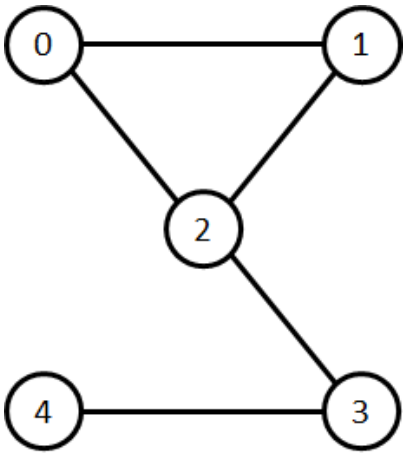
- A: un tableau non-vide de monuments distincts.
- B: un tableau non-vide de monuments distincts.
- A et B doivent être disjoints.
- Cette fonction renvoie `true` s'il y a un monument de A et un monument de B reliés par une route. Sinon, elle renvoie `false`.
- Cette fonction peut être appelée au plus $32\,640$ fois lors de chaque exécution de `longest_trip`, et au plus $150\,000$ fois au total.
- La longueur totale des tableaux A et B passés à cette fonction sur l'ensemble des appels ne peut pas excéder $1\,500\,000$.

L'évaluateur (grader) n'est **pas adaptatif**. Chaque soumission est évaluée sur le même ensemble de tests. Les valeurs de N et D ainsi que les paires de monuments reliées par des routes sont fixés pour chaque appel à `longest_trip` dans chaque test.

Exemples

Exemple 1

Considérons un scénario dans lequel $N = 5$, $D = 1$ et les routes sont illustrées dans la figure suivante :



La fonction `longest_trip` est appelée de la manière suivante :

```
longest_trip(5, 1)
```

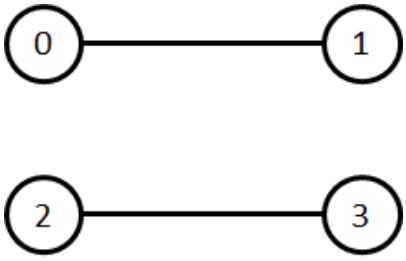
La fonction peut faire des appels à `are_connected` comme ceci :

Appel	Paires reliées par une route	Valeur de retour
`are_connected([0], [1, 2, 4, 3])`	$(0, 1)$ et $(0, 2)$	`true`
`are_connected([2], [0])`	$(2, 0)$	`true`
`are_connected([2], [3])`	$(2, 3)$	`true`
`are_connected([1, 0], [4, 3])`	aucune	`false`

Après le quatrième appel, il s'avère qu'*aucune* des paires $(1, 4)$, $(0, 4)$, $(1, 3)$ et $(0, 3)$ ne sont reliées par une route. Comme la densité du réseau est d'au moins $D = 1$, on voit grâce au triplet $(0, 3, 4)$ que la paire $(3, 4)$ doit être reliée par une route. De manière analogue, les monuments 0 et 1 doivent être reliés.

À ce stade, il est possible de conclure que $t = [1, 0, 2, 3, 4]$ est une excursion de longueur 5 et qu'il n'existe pas d'excursion de longueur strictement plus grande que 5. Par conséquent, la fonction `longest_trip` peut renvoyer $[1, 0, 2, 3, 4]$.

Considérons un autre exemple dans lequel $N = 4$, $D = 1$ et les routes sont illustrées dans la figure suivante :



La fonction `longest_trip` est appelée de la manière suivante :

```
longest_trip(4, 1)
```

Dans ce scénario, la longueur d'une plus grande excursion est 2. Par conséquent, après quelques appels à la fonction `are_connected`, la fonction `longest_trip` peut renvoyer $[0, 1]$, $[1, 0]$, $[2, 3]$ ou $[3, 2]$.

Exemple 2

La sous-tâche 0 contient un test d'exemple additionnel avec $N = 256$ monuments. Ce test est inclus dans l'archive que vous pouvez télécharger sur CMS.

Contraintes

- $3 \leq N \leq 256$
- La somme des N parmi tous les appels à `longest_trip` ne doit pas excéder $1\,024$ dans chaque test.
- $1 \leq D \leq 3$

Sous-tâches

1. (5 points) $D = 3$
2. (10 points) $D = 2$
3. (25 points) $D = 1$. Soit $l^\star$ la longueur d'une plus longue excursion. La fonction `longest_trip` n'a pas à renvoyer une excursion de longueur $l^\star$. À la place, elle doit renvoyer une excursion de longueur d'au moins $\left\lceil \frac{l^\star}{2} \right\rceil$.
4. (60 points) $D = 1$

Dans la sous-tâche 4, votre score est déterminé sur la base du nombre d'appels à la fonction `are_connected` sur une seule invocation de `longest_trip`. Soit q le nombre maximum d'appels parmi toutes les invocations de `longest_trip` de tous les tests de la sous-tâche. Votre score pour cette sous-tâche est calculé selon le tableau suivant :

Condition	Points
$2\,750 < q \leq 32\,640$	20
$550 < q \leq 2\,750$	30
$400 < q \leq 550$	45
$q \leq 400$	60

Si, dans un des tests, les appels à `are_connected` ne sont pas conformes aux contraintes décrites dans Détails d'implémentation ou si le tableau renvoyé par `longest_trip` est incorrect, le score de votre solution pour cette sous-tâche sera 0.

Évaluateur d'exemple (grader)

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

- ligne 1: C

La description des C scénarios suit.

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

- ligne 1: N D
- ligne $1 + i$ ($1 \le i < N$): $U_i[0]$ $U_i[1]$ $\ldots$ $U_i[i-1]$

Ici, chaque U_i ($1 \le i < N$) est un tableau de taille i, décrivant quelles paires de monuments sont reliées par une route. Pour tous i et j tels que $1 \le i < N$ et $0 \le j < i$:

- si les monuments j et i sont reliés par une route, alors la valeur de $U_i[j]$ doit être 1;
- s'il n'y a pas de route reliant les monuments j et i, alors la valeur de $U_i[j]$ doit être 0.

Dans chaque scénario, avant d'appeler `longest_trip`, l'évaluateur d'exemple vérifie si la densité du réseau routier est d'au moins D. Si cette condition n'est pas satisfaite, il affiche le message `Insufficient Density` et termine.

Si l'évaluateur d'exemple détecte une violation du protocole, la sortie de l'évaluateur d'exemple est `Protocol Violation: <MSG>`, où `<MSG>` est l'un des messages d'erreur suivant :

- `invalid array`: dans un appel à `are_connected`, au moins un des tableaux A et B
  - est vide, ou
  - contient un élément qui n'est pas un entier entre 0 et $N-1$ inclus, ou
  - contient le même élément au moins deux fois.
- `non-disjoint arrays`: dans un appel à `are_connected`, les tableaux A et B ne sont pas disjoints.
- `too many calls`: le nombre d'appels faits à `are_connected` excède $32\,640$ sur l'invocation actuelle de `longest trip` ou excède $150\,000$ au total.
- `too many elements`: le nombre total de monuments passés à `are_connected` parmi tous les appels excède $1\,500\,000$.

Sinon, notons $t[0], t[1], \ldots, t[l-1]$ les éléments du tableau renvoyé par `longest_trip` dans un scénario. L'évaluateur d'exemple affiche trois lignes pour ce scénario dans le format suivant :

- ligne 1: l

- ligne 2: $t[0]$ $t[1]$ $\ldots$ $t[l-1]$
- ligne 3: le nombre d'appels à `are_connected` dans ce scénario

Finalement, l'évaluateur d'exemple affiche :

- ligne $1 + 3 \cdot C$: le nombre maximum d'appels à `are_connected` parmi tous les appels à `longest_trip`