



Test de Sélection pour les IOI et EGOI 2025

Xoracle

Limite de temps : 5 secondes

Limite de mémoire : 1024 Mo

Il y a bien longtemps, au cœur des pays Xordiques, vivait un soldat courageux nommé Ronni. Ronni était connu pour son courage et son esprit vif, résolvant souvent des énigmes qui laissaient perplexes même les plus sages. Un jour, Ronni fut convoqué dans la forêt ancestrale où se dressait un arbre mystique. Cet arbre n'était pas ordinaire : il était totalement invisible, ses nœuds et branches cachés aux yeux des mortels. Sur chaque nœud de l'arbre siège un esprit ancien, et le degré de chaque nœud détenait la clé pour comprendre la structure de l'arbre.

L'oracle du royaume, appelé Xoracle, était une entité puissante qui ne pouvait répondre qu'à un seul type de question :

“Donne-moi le XOR bit-à-bit des degrés du nœud A et du nœud B.”

Armé de cette connaissance cryptique, Ronni devait déduire le degré de tous les N nœuds de l'arbre afin de vaincre les esprits anciens et de révéler les secrets de l'arbre. Cependant, le Xoracle ne répondrait qu'à Q requêtes avant de sceller sa sagesse à jamais.

La tâche de Ronni est de déterminer les degrés de tous les nœuds de l'arbre invisible en utilisant les réponses du Xoracle. L'arbre, avec N nœuds et $N - 1$ arêtes, est connexe, ce qui signifie qu'il existe un chemin entre chaque paire de nœuds. Le degré d'un nœud est le nombre d'arêtes qui lui sont connectées. En choisissant stratégiquement des paires de nœuds et en interprétant le XOR bit-à-bit de leurs degrés, Ronni vise à reconstruire les degrés de tous les nœuds de l'arbre.

Interaction

Votre programme doit commencer par lire une ligne contenant deux entiers séparés par un espace N et Q , le nombre de nœuds de l'arbre et le nombre maximum de requêtes que vous pouvez poser au Xoracle. Les nœuds de l'arbre sont numérotés de 1 à N .

Ensuite, votre programme peut effectuer jusqu'à Q requêtes. Pour poser une requête, affichez une ligne de la forme " $? \ i \ j$ " (sans les guillemets) où $1 \leq i, j \leq N$. Le juge répondra alors par un nombre x sur une ligne, où $x = \deg(i) \oplus \deg(j)$. Ici, $\deg(x)$ est le degré du nœud x , et \oplus est le XOR bit-à-bit.

Le XOR bit-à-bit de deux entiers a et b se calcule en regardant leurs représentations binaires. Le i -ème bit du résultat est 1 si et seulement si le i -ème bit de a ou de b vaut 1, mais pas les deux. Cet opérateur est disponible en C++ sous la forme \wedge .

Après vos requêtes, votre programme doit afficher les degrés de tous les nœuds. Pour cela, affichez sur une nouvelle ligne " $!$ " (sans les guillemets) suivi d'un espace puis des N entiers séparés par des espaces, représentant les degrés des N nœuds dans n'importe quel ordre.

Cette opération ne compte pas dans le nombre de requêtes autorisées.

Pour recevoir une réponse à une requête et pour soumettre les degrés à la fin, votre programme doit forcer l'affichage (flush) de la sortie. Cela peut se faire ainsi :

- C++ : `std::cout << std::endl;`

Contraintes

- $2 \leq N \leq 10^5$

Sous-tâches

Sous-tâche	Points	Contraintes
1	8	Le degré maximal de tout nœud est 3. Il existe au moins un nœud de chaque degré 1, 2 et 3. $N \leq 1000$, $Q = N - 1$
2	5	Le degré maximal de tout nœud est 4. Au moins 3 des 4 degrés possibles apparaissent dans l'arbre. $N \leq 1000$, $Q = N - 1$
3	9	$Q = N^2$, $N \leq 300$
4	11	$Q = 35000$, $N \leq 1000$
5	24	$Q = N - 1$, $N \leq 1000$
6	43	$Q = N - 1$

Exemple 1

Observez l'arbre de la Figure 1. Considérez l'interaction suivante, où $>$ désigne le juge et $<$ désigne la solution du candidat.

```
> 4 3
< ? 2 4
> 0
< ? 4 1
> 2
< ? 3 3
> 0
< ! 1 3 1 1
```

1. D'abord, les nombres N et Q sont donnés.
2. Ensuite, une requête pour les nœuds 2 et 4 est faite, donnant le résultat 0.
3. Ensuite, une requête pour les nœuds 4 et 1 est faite, donnant le résultat 2.
4. Ensuite, une requête pour les nœuds 3 et 3 est faite, donnant le résultat 0.
5. Enfin, le programme répond que les degrés de l'arbre sont 1, 3, 1, 1, ce qui est correct.

(L'ordre des degrés n'a pas d'importance.)

Il se peut ou non qu'il soit possible de conclure la bonne réponse avec les requêtes données dans l'exemple.

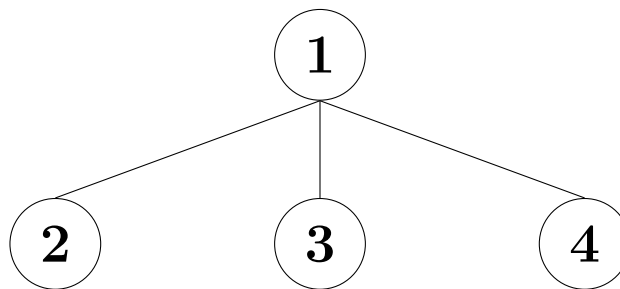


Figure 1: Arbre Exemple 1

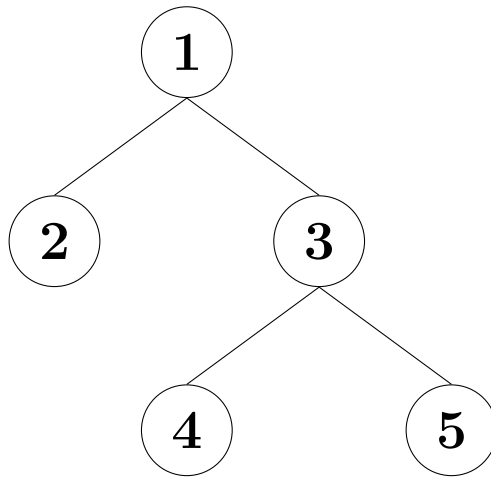


Figure 2: Arbre Exemple 2

Exemple 2

Observez l'arbre de la Figure 2.

Considérez l'interaction suivante, où > désigne le juge et < la solution du candidat.

```

> 5 4
< ? 1 2
> 3
< ? 1 3
> 1
< ? 2 3
> 2
< ? 1 4
> 3
< ! 3 1 1 1 2

```

Il se peut ou non qu'il soit possible de conclure la bonne réponse avec les requêtes données dans l'exemple.