D. Laser Strike

Nom du problème	Laser Strike
Limite de temps	3 secondes
Limite de mémoire	1 gigaoctet

Ann et son amie Kathrin ont récemment découvert un nouveau jeu de société qui est devenu leur jeu préféré : Laser Strike. Dans ce jeu, les deux joueuses collaborent pour retirer N pièces du plateau. Le jeu se déroule en deux phases. La difficulté est que Kathrin n'aura pas des informations complètes sur le jeu. Afin de gagner la partie, Ann et Kathrin doivent collaborer, tout en communiquant le moins possible.

Il y a N pièces uniques sur le plateau, numérotées de 0 à $N-1$. Les deux joueuses peuvent voir ces pièces. Il existe également $N-1$ connexions entre paires de pièces, de sorte qu'il est possible d'atteindre n'importe quelle pièce à partir de n'importe quelle autre pièce en suivant ces connexions. En d'autres termes, ces connexions forment un arbre. **Seule Ann peut voir ces connexions ; Kathrin ne les connaît pas.**

Dans la première phase du jeu, Ann décide d'un ordre $\ell_0, \ell_1, \ldots, \ell_{N-2}$ dans lequel les $N-1$ premières pièces doivent être retirées, jusqu'à ce qu'il n'en reste plus qu'une. Cet ordre sera tenu secret pour Kathrin. Si elle parvient à le reproduire, elles gagneront la partie. Retirer une pièce doit satisfaire la règle suivante : pour pouvoir retirer une pièce, elle doit être connectée avec exactement une pièce restante. En d'autres termes, il doit s'agir d'une feuille de l'arbre formé par les pièces restantes. (Une fois les $N-1$ pièces retirées, la dernière pièce est retirée automatiquement et les joueuses gagnent.) Ann doit choisir un ordre qui correspond à la règle ci-dessus.

Ann écrira également un message à Kathrin, sous la forme d'une chaîne binaire. Ann peut choisir la longueur de ce message, mais plus il est court, plus elles obtiennent de points.

Après cela, la deuxième phase du jeu commence. Le but du jeu est que Kathrin retire $N-1$ pièces du plateau dans l'ordre $\ell_0, \ell_1, \ldots, \ell_{N-2}$. Elle jouera $N-1$ coups. Avant le i-ème coup, Ann indique à Kathrin une paire d'entiers a, b avec les propriétés suivantes :

- $a < b$;
- il reste une paire de pièces directement connectées avec les numéros a et b ; et

- soit a soit b est la bonne pièce ℓ_i qui doit être retirée à ce coup.

Notez que pour Ann, la connexion (a, b) est déterminée de manière unique par la feuille ℓ_i dans l'arbre actuel.

Kathrin retire alors soit a soit b du plateau. Si c'était la bonne pièce (c'est-à-dire ℓ_i) elles continuent à jouer. Sinon, elles perdent la partie.

Votre tâche est d'implémenter les stratégies d'Ann et de Kathrin afin qu'elles gagnent la partie.

Votre programme sera noté en fonction de la longueur du message qu'Ann écrit lors la première phase de jeu.

Implémentation

Il s'agit d'un problème à exécutions mutiples, c'est-à-dire que votre programme sera exécuté deux fois. Lors de la première exécution, il doit implémenter la stratégie de Ann pour la première phase de jeu. Puis, il doit implémenter la stratégie de Kathrin pour la seconde phase de jeu.

La première ligne de l'entrée contient deux entiers, P et N, où P est soit 1 soit 2 (première ou deuxième phase), et N est le nombre de pièces.

La suite de l'entrée dépend de la phase de jeu :

Phase 1 : Ann

Après la première ligne (décrite ci-dessus), les $N-1$ lignes suivantes décrivent l'arbre. Chaque ligne contient deux nombres, a et b ($0 \leq a < b \leq N-1$), indiquant une connexion entre les pièces a et b.

Votre programme doit commencer par générer une chaîne binaire contenant au plus $1\,000$ caractères, chacun étant `0` ou `1`, le message écrit par Ann. Notez que pour générer une chaîne de longueur 0, le programme doit afficher une ligne vide.

Après cela, il doit afficher $N-1$ entiers $\ell_0, \ell_1, \ldots, \ell_{N-2}$ sur des lignes séparées, indiquant l'ordre dans lequel Ann souhaite supprimer les feuilles de l'arbre. L'ordre doit être tel que si les pièces sont retirées une par une de l'arbre dans cet ordre, la pièce retirée doit toujours être une feuille, c'est-à-dire que l'arbre doit toujours rester connecté.

Phase 2 : Kathrin

Après la première ligne (décrite ci-dessus), la ligne suivante contient la chaîne binaire (message d'Ann) de la phase 1.

Après cela, il y aura $N-1$ tours d'interactions, un pour chacun des coups de Kathrin.

Au i-ème coup, votre programme doit commencer par lire deux nombres, a et b ($0 \leq a < b \leq N-1$). L'une de ces pièces est la feuille ℓ_i dans l'ordre d'Ann, et l'autre pièce est la seule pièce restante connectée à ℓ_i. Puis, votre programme doit afficher ℓ_i, indiquant que Kathrin a supprimé cette feuille. Si votre programme n'affiche pas la bonne feuille ℓ_i, les filles perdent la partie et votre réponse sera évalué comme "Wrong Answer" pour ce test.

Détails

Si la *somme* des temps d'exécution des deux exécutions distinctes de votre programme dépasse la limite de temps, votre soumission sera évaluée comme "Time Limit Exceeded".

Assurez-vous de vider la sortie standard après avoir écrit chaque ligne, sinon votre programme pourrait être évalué comme "Time Limit Exceeded". En Python, cela se produit automatiquement lorsque vous utilisez `input()` pour lire les lignes. En C++, `cout << endl;` vide en plus d'afficher une nouvelle ligne ; si vous utilisez printf, utilisez `fflush(stdout);`.

Notez que lire correctement une chaîne vide peut être compliqué. Les templates fournis gèrent correctement ce cas.

Contraintes et scores

- $N = 1\,000$.
- $0 \leq a < b \leq N-1$ pour chaque connexion.

Votre solution sera testée sur un ensemble de sous-tâches, chacune rapportant un certain nombre de points. Chaque sous-tâche contient un ensemble de tests. Pour obtenir les points d'une sous-tâche, vous devez résoudre tous les tests de cette sous-tâche.

Sous-tâche	Score max	Contraintes
1	8	L'arbre est une étoile. Autrement dit, tous les nœuds sauf un sont des feuilles.
2	9	L'arbre est une chaîne. Autrement dit, tous les nœuds, à l'exception de deux nœuds feuilles, ont exactement deux nœuds adjacents.
3	21	L'arbre est une étoile d'où partent des chaînes. Autrement dit, tous les nœuds ont un ou deux nœuds adjacents, sauf un qui en a strictement plus de deux.
4	36	La distance entre deux nœuds est au plus de 10 .
5	26	Aucune contrainte supplémentaire.

Pour chaque sous-tâche que votre programme résout correctement, vous recevrez un score basé sur la formule suivante :

$$\text{score} = S_g \cdot (1 - 0.3 \cdot \log_{10} \max(K, 1)),$$

où S_g est le score maximum pour la sous-tâche, et K est la longueur maximale du message d'Ann sur tous les tests de la sous-tâche. **Votre score pour chaque sous-tâche sera arrondi à l'entier le plus proche.**

Le tableau ci-dessous montre le nombre de points, pour quelques valeurs de K, que votre programme obtiendra s'il résout toutes les sous-tâches avec ce K . En particulier, pour obtenir un score de 100 points, votre solution doit résoudre tous les tests avec $K \leq 1$.

K	1	5	10	50	100	500	1000
Score	100	79	70	49	39	20	11

Outil de test

Pour faciliter le test de votre solution, nous mettons à votre disposition un outil simple que vous pouvez télécharger. Voir "attachments" en bas de la page du problème sur Kattis. L'utilisation de cet outil est facultative. Veuillez noter que le grader officiel sur Kattis est différent de l'outil de test.

Pour utiliser l'outil, créez un fichier d'entrée, tel que "sample1.in", qui doit commencer par un nombre N suivi de $N-1$ lignes décrivant l'arbre, dans le même format que dans la phase 1. Par exemple, pour l'exemple ci-dessous :

```
7
0 1
1 2
2 3
0 4
0 6
1 5
```

Pour les programmes Python, comme par exemple `solution.py` (normalement exécuté avec `pypy3 solution.py`), exécutez :

```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

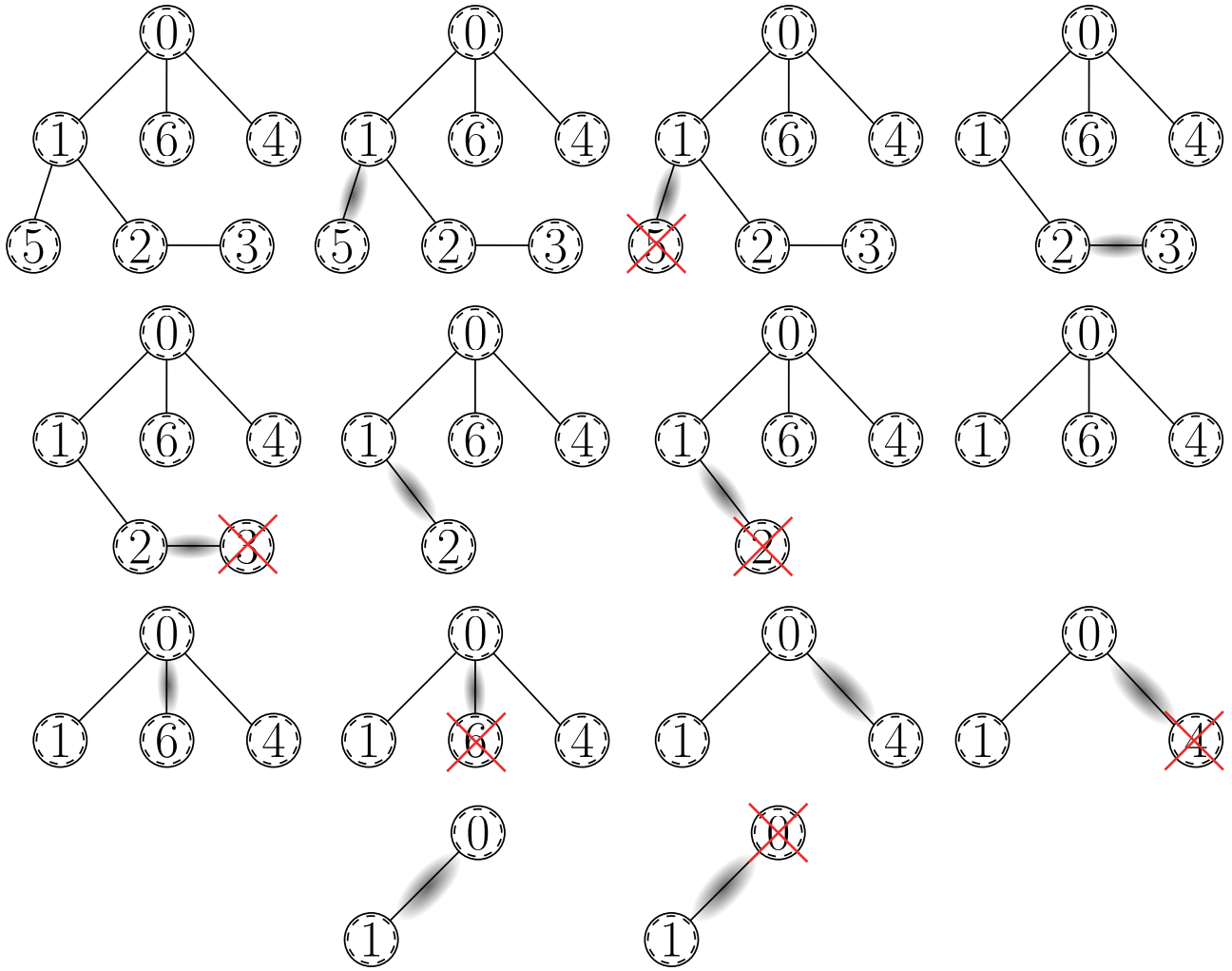
Pour les programmes C++, compilez-les dans un premier temps (par exemple avec `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`) puis exécutez :

```
python3 testing_tool.py ./solution.out < sample1.in
```

Exemple

Notez que l'exemple de cette section a $N = 7$ pour plus de simplicité et ne constitue donc pas un test valide. Votre programme n'a pas à résoudre ce test. Tous les tests officiels auront $N = 1\,000$.

Dans l'exemple, Ann reçoit l'arbre suivant. Dans la première phase, Ann lit l'arbre, sélectionne la chaîne binaire "0110" à envoyer à Kathrin, puis choisit l'ordre $[\ell_0, \ell_1, \ldots \ell_{N-2}] = [5, 3, 2, 6, 4, 0]$ dans lequel les éléments doivent être retirés de l'arbre. Dans la deuxième phase, Kathrin reçoit la chaîne "0110" qui a été envoyée dans la première phase. Elle reçoit alors la paire $(1, 5)$ et décide de supprimer le sommet 5, qui est en effet la feuille. Pour le coup suivant, elle reçoit la paire $(2, 3)$ et retire la feuille 3, et ainsi de suite. Les images suivantes illustrent les interactions :



sortie du grader	votre sortie
1 7	
0 1	
1 2	
2 3	
0 4	
0 6	
1 5	
	0110
	5
	3
	2
	6
	4
	0

sortie du grader	votre sortie
2 7	
0110	
1 5	
	5
2 3	
	3
1 2	
	2
0 6	
	6
0 4	
	4
0 1	
	0