

Multi-ensemble XOR

On vous donne un entier n et $n - 1$ entiers non négatifs a_1, a_2, \dots, a_{n-1} .

Trouver un multiensemble S d'entiers de $\{1, 2, \dots, n - 1\}$ tel que :

- $\sum_{x \in S} x \equiv 0 \pmod{n}$
- $\bigoplus_{x \in S} a_x$ est maximisé, où \bigoplus désigne l'opération XOR bit à bit. L'opérateur XOR bit à bit travaille sur la représentation binaire de deux nombres et effectue l'opération OU exclusif logique sur chaque paire de bits correspondante ; ainsi, 5 (représentation binaire 0101) XOR 3 (représentation binaire 0011) donne 6 (représentation binaire 0110). L'opérateur est `^` en C++, Java et Python.

S'il existe plusieurs multi-ensembles de ce type, vous pouvez renvoyer n'importe lequel d'entre eux.

Détails d'implémentation

Vous devez implémenter la fonction suivante :

```
(int64, int32[]) find_multiset(int32 n, int64[] a)
```

- n : la valeur du modulo
- a : tableau de longueur $n - 1$, où $a[i]$ correspond à a_{i+1}
- La fonction doit renvoyer une paire dont les éléments sont :
 - **Premier élément** : un entier représentant la valeur optimale de $\bigoplus_{x \in S} a_x$ parmi tous les multiensembles valides S
 - **Deuxième élément** : un vecteur représentant un multiensemble optimal S . Les éléments du vecteur doivent être des entiers compris entre 1 et $n - 1$, et la taille de S doit être d'au plus $2n$.

Contraintes

- $1 \leq n \leq 10^5$
- $0 \leq a_i < 2^{62}$ pour chaque $i = 1, 2, \dots, n - 1$

Sous tâches et barème

- **Sous-tâche 1** (20 points) : $n \leq 10$

- **Sous-tâche 2** (40 points) : n est impair
- **Sous-tâche 3** (40 points) : Aucune contrainte supplémentaire

Pour chaque sous-tâche, vous pouvez obtenir un score partiel si votre programme détermine la valeur optimale de $\bigoplus_{x \in S} a_x$ parmi tous les multi-ensembles valides S . Plus précisément, vous obtenez le score total d'une sous-tâche si, dans tous ses cas de test, le premier élément de la paire renvoyée par `find_multiset` est identique à celui renvoyé par l'évaluateur officiel, et si le second élément est un multi-ensemble valide (c'est-à-dire qu'il satisfait aux conditions ci-dessus) qui atteint cette valeur optimale. Vous obtenez 60% du score d'une sous-tâche si, dans tous ses cas de test, le premier élément de la paire renvoyée par `find_multiset` est identique à celui renvoyé par l'évaluateur officiel (quel que soit le second élément), et vous obtenez 0% du score d'une sous-tâche dans le cas contraire.

Exemples

L'appel `find_multiset(3, {5, 10})` suivant devrait renvoyer `{15, {1, 2}}`

- Nous avons $n = 3$ et $a = [5, 10]$ (correspondant à $a_1 = 5, a_2 = 10$).
- Nous devons trouver un multiensemble $S \subseteq \{1, 2\}$ tel que $\sum_{x \in S} x \equiv 0 \pmod{3}$.
- Les multi-ensembles valides incluent : \emptyset (somme = 0), $\{1, 2\}$ (somme = $3 \equiv 0$), $\{1, 1, 1\}$ (somme = $3 \equiv 0$), $\{2, 2, 2\}$ (somme = $6 \equiv 0$), etc.
- Pour $S = \{1, 2\}$: le XOR est $a_1 \oplus a_2 = 5 \oplus 10 = 15$.
- Pour $S = \{1, 1, 1\}$: le XOR est $a_1 \oplus a_1 \oplus a_1 = 5 \oplus 5 \oplus 5 = 5$.
- La valeur XOR maximale est 15, obtenue par $S = \{1, 2\}$.

L'appel `find_multiset(4, {8, 12, 6})` suivant devrait renvoyer `{14, {1, 3}}`

- Nous avons $n = 4$ et $a = [8, 12, 6]$ (correspondant à $a_1 = 8, a_2 = 12, a_3 = 6$).
- Nous devons trouver un multiensemble $S \subseteq \{1, 2, 3\}$ tel que $\sum_{x \in S} x \equiv 0 \pmod{4}$.
- Pour $S = \{1, 3\}$: XOR est $a_1 \oplus a_3 = 8 \oplus 6 = 14$.
- Pour $S = \{2, 2\}$: XOR est $a_2 \oplus a_2 = 12 \oplus 12 = 0$.
- La valeur XOR maximale est 14, obtenue par $S = \{1, 3\}$.

Grader

Le grader lit l'entrée au format suivant :

- Ligne 1 : Un entier n
- Ligne 2 : $n - 1$ entiers a_1, a_2, \dots, a_{n-1}

Le grader appelle `find_multiset(n, a)` et imprime le multiset renvoyé dans le format suivant :

- Première ligne : la valeur renvoyée comme premier élément de la paire
- Deuxième ligne : la taille du multiensemble
- Troisième ligne : les éléments du multiensemble (le cas échéant), séparés par des espaces

Remarque : L'exemple de grader fourni avec ce problème sert uniquement à tester votre solution localement. Le correcteur utilisé pendant le concours peut être différent.