Détection de molécules

Peter travaille dans une société qui a conçu une machine permettant de détecter des molécules. Chaque molécule a un poids entier strictement positif. La machine possède un *intervalle de détection* $[l, u]$, où l et u sont des entiers strictement positifs. Elle peut détecter un ensemble de molécules si et seulement si cet ensemble contient un sous-ensemble de molécules dont le poids total appartient à l'intervalle de détection de la machine.

Formellement, il s'agit de considérer n molécules avec des poids $w_0, \ldots, w_{n-1}$. La détection est réussie s'il existe un ensemble d'indices distincts $I = \{i_1, \ldots, i_m\}$ tel que $l \leq w_{i_1} + \ldots + w_{i_m} \leq u$.

Les caractéristiques de la machine nous garantissent que l'écart entre l et u est supérieur ou égal à celui entre la molécule la plus lourde et la molécule la plus légère. Formellement, $u - l \geq w_{max} - w_{min}$, où $w_{max} = \max(w_0, \ldots, w_{n-1})$ et $w_{min} = \min(w_0, \ldots, w_{n-1})$.

Votre tâche est d'écrire un programme qui permet de trouver un sous-ensemble de molécules avec un poids total appartenant à l'intervalle de détection. Dans le cas contraire, il s'agit de signaler qu'un tel sous-ensemble est inexistant.

Détails de l'implémentation

Vous devez implémenter une fonction:

- `int[] solve(int l, int u, int[] w)`
  - `l` et `u`: les bornes de l'intervalle de détection,
  - `w`: poids des molécules.
  - si le sous-ensemble recherché existe, la fonction doit retourner un tableau d'indices des molécules qui forment un tel sous-ensemble. S'il y a plusieurs réponses correctes, elle doit retourner l'une d'elles.
  - Si le sous-ensemble recherché n'existe pas, la fonction doit retourner un tableau vide.

Pour le langage C, la signature de la fonction est légèrement différente :

- `int solve(int l, int u, int[] w, int n, int[] result)`
  - `n`: le nombre d'éléments dans `w` (c-à-d., le nombre de molécules),
  - les autres paramètres sont identiques à ceux cités précédemment.
  - au lieu de retourner un tableau de m indices (comme précité), la fonction doit écrire les indices dans les m premières cases du tableau `result` et puis

retourner m .
- si le sous-ensemble recherché n'existe pas, la fonction ne doit rien écrire dans le tableau result et doit retourner la valeur 0 .

Votre programme peut écrire les indices dans le tableau retourné (ou le tableau result en C) dans n'importe-quel ordre.

Employez les fichiers modèles fournis pour les détails de l'implémentation relatifs à votre langage de programmation.

Exemples

Exemple 1

`solve(15, 17, [6, 8, 8, 7])`
Dans cet exemple nous disposons de quatre molécules ayant comme poids respectivement 6, 8, 8 et 7. La machine peut détecter des sous-ensembles de molécules de poids total compris entre 15 et 17 inclus. Notez que $17 - 15 \geq 8 - 6$. Le poids total des molécules 1 et 3 est $w_1 + w_3 = 8 + 7 = 15$, donc la fonction peut retourner `[1, 3]`. D'autres réponses correctes sont `[1, 2]` ($w_1 + w_2 = 8 + 8 = 16$) et `[2, 3]` ($w_2 + w_3 = 8 + 7 = 15$).

Exemple 2

`solve(14, 15, [5, 5, 6, 6])`
Dans cet exemple nous disposons de quatre molécules ayant comme poids respectivement 5, 5, 6 et 6, et nous cherchons un sous-ensemble de ces molécules avec un poids total compris entre 14 et 15 inclus. Notez encore une fois que $15 - 14 \geq 6 - 5$. Il n'y a aucun sous-ensemble de molécules de poids total compris entre 14 et 15 donc la fonction doit retourner un tableau vide.

Exemple 3

`solve(10, 20, [15, 17, 16, 18])`
Dans cet exemple nous disposons de quatre molécules ayant comme poids respectivement 15, 17, 16 et 18, et nous cherchons un sous-ensemble de ces molécules avec un poids total compris entre 10 et 20 inclus. Notez encore une fois que $20 - 10 \geq 18 - 15$. Tout sous-ensemble contenant exactement un élément a un poids total entre 10 et 20, donc les réponses correctes possibles sont: `[0]`, `[1]`, `[2]` et `[3]`.

Sous-tâches

1. (9 points): $1 \leq n \leq 100$, $1 \leq w_i \leq 100$, $1 \leq u, l \leq 1000$, tous les w_i sont égaux.
2. (10 points): $1 \leq n \leq 100$, $1 \leq w_i, u, l \leq 1000$, et $\max(w_0, \ldots, w_{n-1}) - \min(w_0, \ldots, w_{n-1}) \leq 1$.

3. (12 points): $1 \leq n \leq 100$ et $1 \leq w_i, u, l \leq 1000$.
4. (15 points): $1 \leq n \leq 10\,000$ et $1 \leq w_i, u, l \leq 10\,000$.
5. (23 points): $1 \leq n \leq 10\,000$ et $1 \leq w_i, u, l \leq 500\,000$.
6. (31 points): $1 \leq n \leq 200\,000$ et $1 \leq w_i, u, l < 2^{31}$.

Évaluateur fourni (grader)

L'évaluateur fourni lit les entrées selon le format suivant :

- ligne 1: les entiers n, l, u.
- ligne 2: n entiers: $w_0, \ldots, w_{n-1}$.