

Detecting Molecules

En Petr treballa per a una companyia que ha construït una màquina per detectar molècules. Cada molècula té un pes enter positiu. La màquina té un *interval de detecció* $[l, u]$, on l i u són enters positius. La màquina pot detectar un conjunt de molècules si i només si aquest conjunt conté un subconjunt de molècules el pes total del qual pertany a l'interval de detecció de la màquina.

Formalment, considereu n molècules amb pesos w_0, \dots, w_{n-1} . La detecció té èxit si hi ha un conjunt d'índexs $I = \{i_1, \dots, i_m\}$ tal que $l \leq w_{i_1} + \dots + w_{i_m} \leq u$.

Les especificacions de la màquina garanteixen que la diferència entre l i u és més gran o igual que la diferència entre el pes de la molècula més pesada i el de la més lleugera. Formalment, $u - l \geq w_{max} - w_{min}$, on $w_{max} = \max(w_0, \dots, w_{n-1})$ i $w_{min} = \min(w_0, \dots, w_{n-1})$.

La teva tasca consisteix a escriure un programa que trobi qualsevol subconjunt de molècules el pes total del qual es trobi dins de l'interval de detecció, o bé que determini que no n'hi ha cap.

Detalls d'implementació

Se us demana que implementeu una funció (mètode):

- `int[] solve(int l, int u, int[] w)`
 - l i u : els extrems de l'interval de detecció,
 - w : pes de les molècules.
 - si el subconjunt que es demana existeix, la funció ha de retornar un array d'índexs de molècules que formi qualsevol d'aquests subconjunts. Si hi ha més d'una resposta correcta, en podeu retornar qualsevol.
 - si el subconjunt que es demana no existeix, la funció ha de retornar un array buit.

La capçalera de la funció és lleugerament diferent en el llenguatge C:

- `int solve(int l, int u, int[] w, int n, int[] result)`
 - n : el nombre d'elements que hi ha a w (és a dir, el nombre de molècules),
 - la resta de paràmetres són els mateixos que abans.
 - en comptes de retornar un array de m índexs (com abans), la funció ha de desar els índexs a les primeres m posicions de l'array `result`, i a continuació retornar m .
 - si el subconjunt que es demana no existeix, la funció no ha de desar res a

l'array `result` i ha de retornar `0`.

El vostre programa pot escriure els índexs dins l'array que retorna (o l'array `result` en C) en qualsevol ordre.

Si us plau, feu servir els arxius de mostra per veure els detalls de la implementació al vostre llenguatge de programació.

Exemples

Exemple 1

`solve(15, 17, [6, 8, 8, 7])`

En aquest exemple tenim quatre molècules amb pesos 6, 8, 8 i 7. La màquina pot detectar subconjunts de molècules si el pes total es troba entre 15 i 17, ambdós inclosos. Fixeu-vos que $17 - 15 \geq 8 - 6$. El pes total de les molècules 1 i 3 és

$w_1 + w_3 = 8 + 7 = 15$, de manera que la funció pot retornar `[1, 3]`. Les respostes `[1, 2]` ($w_1 + w_2 = 8 + 8 = 16$) i `[2, 3]` ($w_2 + w_3 = 8 + 7 = 15$) també serien correctes.

Exemple 2

`solve(14, 15, [5, 5, 6, 6])`

En aquest exemple tenim quatre molècules amb pesos 5, 5, 6 i 6, i en busquem un subconjunt amb pes total entre 14 i 15 inclosos. Novament, fixeu-vos que

$15 - 14 \geq 6 - 5$. No hi ha cap subconjunt de molècules amb pes total entre 14 i 15, de manera que la funció ha de retornar un array buit.

Exemple 3

`solve(10, 20, [15, 17, 16, 18])`

En aquest exemple tenim quatre molècules amb pesos 15, 17, 16 i 18, i en busquem un subconjunt amb pes total entre 10 i 20 inclosos. Novament, fixeu-vos que

$20 - 10 \geq 18 - 15$. Qualsevol subconjunt que contingui un sol element tindrà pes total entre 10 i 20, de manera que les solucions correctes possibles són: `[0]`, `[1]`, `[2]` i `[3]`.

Subtasques

- (9 punts): $1 \leq n \leq 100$, $1 \leq w_i \leq 100$, $1 \leq u, l \leq 1000$ tots els valors w_i són iguals.
- (10 punts): $1 \leq n \leq 100$, $1 \leq w_i, u, l \leq 1000$, i $\max(w_0, \dots, w_{n-1}) - \min(w_0, \dots, w_{n-1}) \leq 1$.
- (12 punts): $1 \leq n \leq 100$ i $1 \leq w_i, u, l \leq 1000$.
- (15 punts): $1 \leq n \leq 10000$ i $1 \leq w_i, u, l \leq 10000$.
- (23 punts): $1 \leq n \leq 10000$ i $1 \leq w_i, u, l \leq 500000$.
- (31 punts): $1 \leq n \leq 200000$ i $1 \leq w_i, u, l < 2^{31}$.

Grader de mostra

El grader de mostra llegeix l'entrada en el format següent:

- línia 1: enters n , l , u .
- línia 2: n enters: w_0, \dots, w_{n-1} .