



Moleculen detecteren

Peter werkt voor een bedrijf dat een machine heeft gemaakt om moleculen te kunnen detecteren. Elke molecule heeft een positief integer gewicht. De machine heeft een *detectiebereik* $[l, u]$, waarbij l en u positieve integers zijn. De machine kan een verzameling moleculen enkel en alleen detecteren als deze verzameling een deelverzameling aan moleculen bevat waarvan het totale gewicht binnen het bereik ligt van de machine.

Formeel: beschouw n moleculen met gewichten w_0, \dots, w_{n-1} . De detectie is succesvol als er een verzameling van verschillende indices $I = \{i_1, \dots, i_m\}$ bestaat, zodat $l \leq w_{i_1} + \dots + w_{i_m} \leq u$

Dankzij de specificaties van de machine, is de opening tussen l en u gegarandeerd groter dan of gelijk aan het verschil in gewicht van de zwaarste en de lichtste molecule. Formeel: $u - l \geq w_{max} - w_{min}$, waar $w_{max} = \max(w_0, \dots, w_{n-1})$ en $w_{min} = \min(w_0, \dots, w_{n-1})$.

Jij moet een programma schrijven dat of één of andere deelverzameling van de moleculen vindt waarvan het totale gewicht binnen het detectiebereik valt, of een programma dat vaststelt dat er niet zo'n deelverzameling bestaat.

Implementatie details

Jij moet één functie (methode) implementeren:

- `int[] solve(int l, int u, int[] w)`
 - l en u : de uiteinden van het detectiebereik,
 - w : de gewichten van de moleculen.
 - als de gevraagde deelverzameling bestaat, geeft de functie een array van indices van de moleculen in één van de mogelijke deelverzamelingen terug. Als er meerdere correcte antwoorden zijn, moet je er één naar keuze terug geven.
 - als de gevraagde deelverzameling niet bestaat, geeft de functie een lege array terug.

Voor de taal C is de beschrijving van de functie enigszins afwijkend:

- `int solve(int l, int u, int[] w, int n, int[] result)`
 - n : het aantal elementen in w (dat wil zeggen het aantal moleculen),
 - de andere parameters zijn hetzelfde als hierboven beschreven.
 - in plaats van het terug geven van een array van m indices (zoals

hierboven), moet de functie de indices in de eerste m cellen van array `result` wegschrijven en dan m terug geven.

- als de gevraagde deelverzameling niet bestaat, moet de functie niets in de array `result` schrijven en de waarde `0` terug geven.

Je programma mag de indices in willekeurige volgorde in de array die moet worden terug gegeven, plaatsen (of in de `result` array in C).

Gelieve de voorziene template bestanden te gebruiken voor de details van de implementatie in jouw programmeertaal.

Voorbeelden

Voorbeeld 1

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

In dit voorbeeld zijn er vier moleculen met gewichten 6, 8, 8 en 7. De machine kan deelverzamelingen van moleculen detecteren als hun totale gewicht tussen 15 en 17 ligt (grenzen inclusief). Merk op dat $17 - 15 \geq 8 - 6$. Het totale gewicht van de moleculen 1 en 3 is $w_1 + w_3 = 8 + 7 = 15$, dus de functie kan `[1, 3]` terug geven. Andere mogelijke correcte antwoorden zijn `[1, 2]` ($w_1 + w_2 = 8 + 8 = 16$) en `[2, 3]` ($w_2 + w_3 = 8 + 7 = 15$).

Voorbeeld 2

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

In dit voorbeeld zijn er vier moleculen met gewicht 5, 5, 6 en 6, en we zoeken naar een deelverzameling met een totaal gewicht tussen 14 en 15 (grenzen inclusief). Merk weer op dat $15 - 14 \geq 6 - 5$. Er is geen deelverzameling moleculen met een totaalgewicht tussen 14 en 15. Daarom moet de functie een lege array terug geven.

Voorbeeld 3

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

In dit voorbeeld hebben we vier moleculen met gewichten 15, 17, 16 en 18, en we zoeken naar een deelverzameling met een totaalgewicht tussen 10 en 20 (grenzen inclusief). Merk opnieuw op dat $20 - 10 \geq 18 - 15$. Elke deelverzameling van precies één element heeft een gewicht tussen 10 en 20, dus de mogelijke correcte antwoorden zijn: `[0]`, `[1]`, `[2]` en `[3]`.

Subtasks

- (9 punten): $1 \leq n \leq 100$, $1 \leq u, l \leq 1000$, alle w_i zijn gelijk.
- (10 punten): $1 \leq n \leq 100$, $1 \leq w_i, u, l \leq 1000$, en $\max(w_0, \dots, w_{n-1}) - \min(w_0, \dots, w_{n-1}) \leq 1$.
- (12 punten): $1 \leq n \leq 100$ en $1 \leq w_i, u, l \leq 1000$.

4. (15 punten): $1 \leq n \leq 10\,000$ en $1 \leq w_i, u, l \leq 10\,000$.
5. (23 punten): $1 \leq n \leq 10\,000$ en $1 \leq w_i, u, l \leq 500\,000$
6. (31 punten): $1 \leq n \leq 200\,000$ en $1 \leq w_i, u, l < 2^{31}$.

Voorbeeld grader

De voorbeeld grader leest de invoer in het volgende formaat:

- regel 1: integers n, l, u .
- regel 2: n integers: w_0, \dots, w_{n-1} .