



Detecting Molecules

Петър работи за компания, която строи машина за откриване на молекули. Всяка молекула има положително целочислено тегло. Машината има обхват за откриване $[l, u]$, където l и u са положителни цели числа. Машината може да открие множество от молекули тогава и само тогава, когато това множество съдържа подмножество от молекули с общо тегло, което в обхвата на машината.

По-точно казано, да разгледаме n молекули с положителни целочислени тегла w_0, \dots, w_{n-1} . Откриването е успешно, ако съществува множество от различни индекси $I = \{i_1, \dots, i_m\}$, такива че $l \leq w_{i_1} + \dots + w_{i_m} \leq u$.

Поради спецификата на машината, разликата между u и l е гарантирано по-голяма (или равна) на разликата между теглата на най-тежката и най-леката молекула. Формално казано, $u - l \geq w_{\max} - w_{\min}$, където $w_{\max} = \max(w_0, \dots, w_{n-1})$ и $w_{\min} = \min(w_0, \dots, w_{n-1})$.

Вашата задача е да напишете програма, която намира кое да е множество от молекули с общо тегло, което е в обхвата на машината или открива, че не съществува такова подмножество.

Детайли по реализацията:

Трябва да реализирате една функция:

- `int[] solve(int l, int u, int[] w)`
 - l и u са краищата на обхвата на машината,
 - w : теглата на молекулите.
 - ако търсеното подмножество съществува, функцията трябва да върне един масив от индекси на молекули, които образуват едно такова подмножество. Ако има няколко верни отговора, върнете един от тях.
 - ако търсеното подмножество не съществува, функцията трябва да върне празен масив.

За езика C функцията изглежда малко по-различно:

- `int solve(int l, int u, int[] w, int n, int[] result)`
 - n : брой на елементите в w (т.е. брой на молекулите),
 - останалите параметри са както по-горе.
 - вместо връщане на масив от m индекса (както е по-горе), функцията трябва да запише индексите в първите m клетки на масива `result` и тогава да върне m .

- о ако търсеното подмножество не съществува, функцията не трябва да записва нищо в масива `result` и трябва да върне `0`.

Примери

Пример 1

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

В този пример имаме 4 молекули с тегла 6, 8, 8 и 7. Машината може да открие помножество от молекули с тегла между 15 и 17, включително. Отбелязваме, че $17-15 \geq 8-6$. Общото тегло на молекули 1 и 3 е $w_1 + w_3 = 8 + 7 = 15$, така че функцията може да върне `[1, 3]`. Други възможни правилни отговори са `[1, 2]` ($w_1 + w_2 = 8 + 8 = 16$) и `[2, 3]` ($w_2 + w_3 = 8 + 7 = 15$).

Пример 2

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

В този пример имаме 4 молекули с тегла 5, 5, 6 и 6, и търсим подмножество с общо тегло между 14 и 15, включително. Отбелязваме, че $15-14 \geq 6-5$. Сега няма подмножество от молекули с общо тегло между 14 и 15 така, че функцията трябва да върне празен масив.

Пример 3

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

В този пример имаме 4 молекули с тегла 15, 17, 16 и 18, и търсим подмножество с общо тегло между 10 и 20, включително. Отбелязваме, че $20-10 \geq 18-15$. Всяко подмножество състоящо се от точно един елемент удовлетворява изискванията, така че правилни отговори са: `[0]`, `[1]`, `[2]` и `[3]`.

Подзадачи

- (9 точки): $1 \leq n \leq 100$, $1 \leq w_i \leq 100$, $1 \leq u, l \leq 1000$ и всички w_i са равни.
- (10 точки): $1 \leq n \leq 100$, $1 \leq w_i, u, l \leq 1000$, и $\max(w_0, \dots, w_{n-1}) - \min(w_0, \dots, w_{n-1}) \leq 1$.
- (12 точки): $1 \leq n \leq 100$ и $w_{i,u,l} \leq 1000$.
- (15 точки): $n \leq 10,000$ и $w_{i,u,l} \leq 10,000$.
- (23 точки): $1 \leq n \leq 10,000$ и $w_{i,u,l} \leq 500,000$.
- (31 точки): $n \leq 200,000$ и $w_{i,u,l} < 2^{31}$.

Примерен грейдер

Примерният грейдер чете входа в следния формат:

- ред 1: цели числа n , l , u .
- ред 2: n цели числа: w_0, \dots, w_{n-1} .