Wykrywacz cząsteczek (Detecting Molecules)

Firma, w której pracuje Petr, skonstruowała maszynę do wykrywania cząsteczek. Każda cząsteczka ma masę wyrażającą się dodatnią liczbą całkowitą. Maszyna ma określony *zakres pomiarowy* $[l, u]$, gdzie l i u są dodatnimi liczbami całkowitymi. Maszyna może wykryć zbiór cząsteczek wtedy i tylko wtedy, gdy zbiór ten zawiera podzbiór, którego łączna masa należy do zakresu pomiarowego maszyny.

Formalnie, rozważmy n cząsteczek o masach $w_0, \ldots, w_{n-1}$. Proces wykrywania kończy się powodzeniem, jeśli istnieje zbiór parami różnych indeksów $I = i_1, \ldots, i_m$ taki że $l \leq w_{i_1} + \ldots + w_{i_m} \leq u$.

Konstrukcja maszyny gwarantuje, że różnica między l i u jest nie mniejsza niż różnica mas najcięższej i najlżejszej cząsteczki. Formalnie, $u - l \geq w_{max} - w_{min}$, gdzie $w_{max} = \max(w_0, \ldots, w_{n-1})$ i $w_{min} = \min(w_0, \ldots, w_{n-1})$.

Twoim zadaniem jest napisanie programu, który albo wyznaczy jakikolwiek podzbiór zbioru cząsteczek, którego łączna masa należy do zakresu pomiarowego maszyny, albo stwierdzi, że taki podzbiór nie istnieje.

Szczegóły implementacji

Powinieneś napisać jedną funkcję (metodę):

- `int[] solve(int l, int u, int[] w)`
  - `l` i `u`: końce zakresu pomiarowego,
  - `w`: masy cząsteczek.
  - Jeśli żądany podzbiór istnieje, funkcja powinna zwrócić tablicę indeksów cząsteczek, które tworzą dowolny taki podzbiór. Jeśli jest więcej niż jedna poprawna odpowiedź, wynikiem funkcji może być dowolna z nich.
  - Jeśli żądany podzbiór nie istnieje, funkcja powinna zwrócić pustą tablicę.

W języku C sygnatura funkcji jest minimalnie inna:

- `int solve(int l, int u, int[] w, int n, int[] result)`
  - `n`: liczba elementów tablicy `w` (tj. liczba cząsteczek),
  - pozostałe parametry są takie same jak powyżej.
  - Zamiast zwracać tablicę opisującą m indeksów (jak powyżej), funkcja powinna zapisać te indeksy do pierwszych m komórek tablicy `result` i zwrócić m.
  - Jeśli żądany podzbiór nie istnieje, funkcja nie powinna niczego zapisywać do tablicy `result` i powinna zwrócić 0.

Twój program może zapisać indeksy do zwracanej tablicy (lub do tablicy `result` w przypadku języka C) w dowolnej kolejności.

Szczegóły implementacji w Twoim języku programowania znajdują się w dostarczonych plikach z szablonami.

Przykłady

Przykład 1

`solve(15, 17, [6, 8, 8, 7])`
W tym przykładzie mamy cztery cząsteczki o masach 6, 8, 8 i 7. Maszyna potrafi wykrywać podzbiory cząsteczek o łącznej masie między 15 a 17 włącznie. Zauważ, że $17 - 15 \geq 8 - 6$. Łączna masa cząsteczek 1 i 3 to $w_1 + w_3 = 8 + 7 = 15$, tak więc funkcja może zwrócić `[1, 3]`. Inne poprawne odpowiedzi to `[1, 2]` ($w_1 + w_2 = 8 + 8 = 16$) i `[2, 3]` ($w_2 + w_3 = 8 + 7 = 15$).

Przykład 2

`solve(14, 15, [5, 5, 6, 6])`
W tym przykładzie mamy cztery cząsteczki o masach 5, 5, 6 i 6 i szukamy podzbioru o łącznej masie między 14 a 15 włącznie. Znów, zauważ że $15 - 14 \geq 6 - 5$. W tym przypadku nie ma żadnego podzbioru cząsteczek o łącznej masie między 14 a 15, więc wynikiem funkcji powinna być pusta tablica.

Przykład 3

`solve(10, 20, [15, 17, 16, 18])`
W tym przykładzie mamy cztery cząsteczki o masach 15, 17, 16 i 18 i szukamy podzbioru o łącznej masie między 10 a 20 włącznie. Znów, zauważ że $20 - 10 \geq 18 - 15$. Każdy podzbiór jednoelementowy ma łączną masę między 10 a 20, tak więc możliwe poprawne wyniki to: `[0]`, `[1]`, `[2]` i `[3]`.

Podzadania

1. (9 punktów): $1 \leq n \leq 100$, $1 \leq w_i \leq 100$, $1 \leq u, l \leq 1000$, wszystkie w_i są równe.
2. (10 punktów): $1 \leq n \leq 100$, $1 \leq w_i, u, l \leq 1000$ i $\max(w_0, \ldots, w_{n-1}) - \min(w_0, \ldots, w_{n-1}) \leq 1$.
3. (12 punktów): $1 \leq n \leq 100$ i $1 \leq w_i, u, l \leq 1000$.
4. (15 punktów): $1 \leq n \leq 10\,000$ i $1 \leq w_i, u, l \leq 10\,000$.
5. (23 punkty): $1 \leq n \leq 10\,000$ i $1 \leq w_i, u, l \leq 500\,000$.
6. (31 punktów): $1 \leq n \leq 200\,000$ i $1 \leq w_i, u, l < 2^{31}$.

Przykładowy program sprawdzający

Przykładowy program sprawdzający wczytuje dane w następującym formacie:

- wiersz 1: liczby całkowite n, l, u.
- wiersz 2: n liczb całkowitych: $w_0, \ldots, w_{n-1}$.