



Detecting Molecules

Petr está trabajando para una compañía que ha construido una máquina para identificar moléculas. Cada molécula tiene un peso entero positivo. Esta máquina tiene un *rango de identificación* $[l, u]$, donde l y u son enteros positivos. La máquina puede identificar un conjunto de moléculas si este conjunto contiene un subconjunto de moléculas cuyo peso total pertenece al rango de identificación de la máquina.

Formalmente, considere n moléculas con pesos w_0, \dots, w_{n-1} . La identificación es exitosa si existe un conjunto de índices diferentes $I = \{i_1, \dots, i_m\}$ tal que $l \leq w_{i_1} + \dots + w_{i_m} \leq u$.

Dadas las especificaciones de la máquina, se garantiza que la brecha entre l y u será mayor o igual que la brecha entre los pesos de la molécula más pesada y la molécula menos pesada. Formalmente, $u - l \geq w_{max} - w_{min}$, donde $w_{max} = \max(w_0, \dots, w_{n-1})$ y $w_{min} = \min(w_0, \dots, w_{n-1})$.

Tu tarea es escribir un programa que o bien encuentre cualquier subconjunto de moléculas con peso total en el rango de identificación, o bien determine que tal subconjunto no existe.

Detalles de Implementación

Debes implementar una función (método):

- `int[] solve(int l, int u, int[] w)`
 - `l` y `u`: los extremos del rango del rango de detección,
 - `w`: pesos de las moléculas.
 - Si existen varios subconjuntos válidos, la función debe retornar un arreglo con los índices de las moléculas miembro de cualquier subconjunto válido. Si hubieran varias soluciones, retorne cualquiera de ellas.
 - Si el subconjunto pedido no existe, la función debe retornar un arreglo vacío.

Para el lenguaje C las definiciones de funciones difieren ligeramente:

- `int solve(int l, int u, int[] w, int n, int[] result)`
 - `n`: número de elementos en `w` (esto es, el número de moléculas),
 - los otros parámetros son iguales que la definición previa.
 - en vez de retornar un arreglo de m índices (como arriba), la función debe escribir los índices en las primeras m celdas del arreglo `result` y luego

retornar m .

- Si el subconjunto solicitado no existe, la función debe escribir nada en `result` y retornar `0`.

Tu programa debe escribir los índices en el arreglo retornado (o hacia `result`) resultYour program may write the indices into the returned array (or to the `result` array in C) in any order.

Por favor, utiliza los archivos plantillas provistos para los detalles de implementación en tu lenguaje de programación.

Ejemplos

Ejemplo 1

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

En este ejemplo tenemos cuatro moléculas con pesos 6, 8, 8 y 7. La máquina puede identificar los subconjuntos de moléculas con peso total entre 15 y 17, inclusive. Note, que $17 - 15 \geq 8 - 6$. El peso total The total weight of molecules 1 and 3 is

$w_1 + w_3 = 8 + 7 = 15$, so the function can return `[1, 3]`. Other possible correct answers are `[1, 2]`

($w_1 + w_2 = 8 + 8 = 16$) and `[2, 3]` ($w_2 + w_3 = 8 + 7 = 15$).

Example 2

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

In this example we have four molecules with weights 5, 5, 6 and 6, and we are looking for a subset of them with total weight between 14 and 15, inclusive. Again, note that

$15 - 14 \geq 6 - 5$. There is no subset of molecules with total weight between 14 and 15 so the function should return an empty array.

Example 3

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

In this example we have four molecules with weights 15, 17, 16 and 18, and we are looking for a subset of them with total weight between 10 and 20, inclusive. Again, note that $20 - 10 \geq 18 - 15$. Any subset consisting of exactly one element has total weight between 10 and 20, so the possible correct answers are: `[0]`, `[1]`, `[2]` and `[3]`.

Subtasks

- (9 points): $1 \leq n \leq 100$, $1 \leq w_i \leq 100$, all w_i are equal.
- (10 points): $1 \leq n \leq 100$, $1 \leq w_i \leq 1000$, and $\max(w_0, \dots, w_{n-1}) - \min(w_0, \dots, w_{n-1}) \leq 1$.
- (12 points): $1 \leq n \leq 100$ and $1 \leq w_i, u, l \leq 1000$.
- (15 points): $1 \leq n \leq 10000$ and $1 \leq w_i, u, l \leq 10000$.
- (23 points): $1 \leq n \leq 10000$ and $1 \leq w_i, u, l \leq 500000$.

6. (31 points): $1 \leq n \leq 200\,000$ and $1 \leq w_i, u, l < 2^{31}$.

Sample grader

The sample grader reads the input in the following format:

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