Определение молекул

Петр работает в компании, которая создала машину для обнаружения молекул. Каждая молекула имеет положительный целочисленный вес. Машина имеет интервал обнаружения $[l, u]$, где l и u являются положительными целыми числами. Машина может обнаружить множество молекул, если и только если оно содержит подмножество молекул с общим весом, принадлежащим интервалу определения машины.

Рассмотрим, формально, n молекул с целыми положительными весами $w_0, \ldots, w_{n-1}$. Обнаружение молекул является успешным, если существует множество различных индексов $I = i_1, \ldots, i_m$, таких, что $l \leq w_{i_1} + \ldots w_{i_m} \leq u$. Из-за особенностей машины, разница между l и u гарантированно будет больше или равна разнице весов между самой тяжелой и самой легкой молекулами.
Формально, $u - l \geq w_{max} - w_{min}$, где $w_{max} = \max(w_0, \ldots, w_{n-1})$ и $w_{min} = \min(w_0, \ldots, w_{n-1})$.

Ваша задача - написать программу, которая либо находит какую-то одну подгруппу молекул с общим весом в пределах диапазона обнаружения, или определяет, что не существует такого подмножества.

Детали реализации

Вы должны написать одну функцию (метод):

- `int[] solve(int l, int u, int[] w)`
  - `l` и `u`: границы диапазона обнаружения,
  - `w`: веса молекул.
  - Если требуемое подмножество существует, то функция должна возвращать массив индексов молекул, которые образуют какую-то одну такую подгруппу. Если есть несколько правильных ответов, то возвращать любой из них.
  - Если требуемое подмножество не существует, то функция должна возвращать пустой массив.

Для языка C функция пишется немного иначе:

- `int solve(int l, int u, int[] w, int n, int[] result)`
  - `n`: количество элементов в `w` (т.е., число молекул),
  - остальные параметры такие же, что и выше.
  - Вместо возвращения массива m индексов (как описано выше),

функция должна записать индексы в первые m элементов массива result, а затем вернуть m.
- Если требуемое подмножество не существует, то функция не должна ничего записывать в массив result и должна возвратить 0.

Ваша программа может записать индексы в возвращаемый массив (или в массив result в C) в любом порядке.

Пожалуйста, используйте предоставленные файлы шаблонов для деталей реализации на вашем языке программирования.

Примеры

Пример 1

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

В этом примере у нас есть четыре молекулы с весами 6, 8, 8 и 7. Машина может обнаружить подмножества молекул с общим весом от 15 до 17 лет включительно. Заметим, что $17 - 15 \geq 8 - 6$. Общий вес молекул 1 и 3 $w_1 + w_3 = 8 + 7 = 15$, так что функция может возвратить [1, 3]. Другие возможные правильные ответы [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 этом примере мы имеем четыре молекулы с весом 5, 5, 6 и 6, и мы ищем подмножества из них с общим весом от 14 до 15 включительно. Опять же, обратите внимание, что $15 - 14 \geq 6 - 5$. Нет подмножества молекул с общим весом между 14 и 15, так что функция должна возвращать пустой массив.

Пример 3

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

В этом примере мы имеем четыре молекулы с весами 15, 17, 16 и 18, и ищем подмножества из них с общим весом от 10 до 20 включительно. Опять же, обратите внимание, что $20 - 10 \geq 18 - 15$. Любое подмножество, состоящее из одного элемента точно удовлетворяет требованию, так что правильные ответы: [0], [1], [2] и [3].

Подзадачи

1. (9 баллов): $n \leq 100$, $w_i \leq 100$, все w_i равны.
2. (10 баллов): $n \leq 100, w_i \leq 1000$, и $max(w_0, \ldots, w_{n-1}) - min(w_0, \ldots, w_{n-1}) \leq 1$
3. (12 баллов): $n \leq 100$ и $w_i, u, l \leq 1000$.
4. (15 баллов): $n \leq 10\,000$ и $w_i, u, l \leq 10\,000$.
5. (23 баллов): $n \leq 10\,000$ и $w_i, u, l \leq 500\,000$

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

Пример проверяющего модуля

Проверяющий модуль получает данные в следующем формате:

- Строка 1: целые n , l , u .
- Строка 2: n целые: $w_0, \ldots, w_{n-1}$.