

Обнаружение молекул

Петр работает в компании, которая создала прибор для обнаружения молекул. Каждая молекула имеет целый положительный вес. Прибор характеризуется интервалом обнаружения $[l, u]$, где l и u целые положительные числа. Прибор может обнаружить множество молекул тогда и только тогда, когда это множество содержит такое подмножество, что суммарный вес молекул в нем принадлежит интервалу обнаружения прибора.

Более формально, рассмотрим n молекул с весами w_0, \dots, w_{n-1} . Обнаружение считается успешным, если существует множество различных индексов

$$I = \{i_1, \dots, i_m\} \text{ такое, что } l \leq w_{i_1} + \dots + w_{i_m} \leq u.$$

В силу особенностей работы прибора разница между l и u гарантированно больше либо равна разнице весов между самой тяжелой и самой легкой молекулами. Более формально, $u - l \geq w_{max} - w_{min}$, где $w_{max} = \max(w_0, \dots, w_{n-1})$ и $w_{min} = \min(w_0, \dots, w_{n-1})$.

Требуется написать программу, которая либо находит любое подмножество молекул с суммарным весом, принадлежащим интервалу обнаружения прибора, либо определяет, что такого подмножества не существует.

Детали реализации

Вам следует реализовать одну функцию (метод):

- `int[] solve(int l, int u, int[] w)`
 - l и u : границы интервала обнаружения,
 - w : веса молекул.
 - Если требуемое подмножество существует, то функция должна вернуть массив индексов молекул, которые формируют любое такое подмножество. Если существует несколько правильных ответов, верните любой из них.
 - Если требуемого подмножества не существует, то функция должна вернуть пустой массив.

Для языка программирования C сигнатура функции немного отличается:

- `int solve(int l, int u, int[] w, int n, int[] result)`
 - n : количество элементов в w (то есть число молекул),
 - остальные параметры такие же, как описано выше.
 - Вместо того, чтобы возвращать массив состоящий из m индексов (как указано выше), функция должна записать индексы в первые m ячеек

массива `result` и затем вернуть `m`.

- Если требуемого подмножества не существует, то функция должна вернуть `0`, не записывая ничего в массив `result`.

Ваша программа может записывать индексы в возвращаемый массив (или в массив `result` для языка C) в любом порядке.

Пожалуйста, используйте предоставленные шаблоны файлов для уточнения реализации на выбранном вами языке программирования.

Примеры

Пример 1

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

В этом примере есть четыре молекулы с весами 6, 8, 8 и 7. Прибор может обнаружить подмножества молекул с суммарным весом от 15 до 17 включительно. Обратите внимание, что $17 - 15 \geq 8 - 6$. Суммарный вес молекул 1 и 3 равен $w_1 + w_3 = 8 + 7 = 15$, таким образом функция может вернуть `[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 есть четыре молекулы с весами 5, 5, 6 и 6. Требуется найти подмножество с суммарным весом от 14 до 15 включительно. Опять же, обратите внимание, что $15 - 14 \geq 6 - 5$. Для данного примера не существует подмножества молекул с суммарным весом от 14 до 15, соответственно функция должна вернуть пустой массив.

Пример 3

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

В этом примере есть четыре молекулы с весами 15, 17, 16 и 18. Требуется найти подмножество с суммарным весом от 10 до 20 включительно. Вновь, обратите внимание, что $20 - 10 \geq 18 - 15$. Любое подмножество, состоящее из одного элемента, имеет вес от 10 до 20, соответственно возможные правильные ответы это `[0]`, `[1]`, `[2]` и `[3]`.

Система оценивания

1. (9 баллов): $1 \leq n \leq 100$, $1 \leq w_i \leq 100$, $1 \leq u, l \leq 1000$, все w_i равны.
2. (10 баллов): $1 \leq n \leq 100$, $1 \leq w_i, u, l \leq 1000$, и $\max(w_0, \dots, w_{n-1}) - \min(w_0, \dots, w_{n-1}) \leq 1$.
3. (12 баллов): $1 \leq n \leq 100$ и $1 \leq w_i, u, l \leq 1000$.
4. (15 баллов): $1 \leq n \leq 10000$ и $1 \leq w_i, u, l \leq 10000$.

5. (23 балла): $1 \leq n \leq 10\,000$ и $1 \leq w_i, u, l \leq 500\,000$.

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

Пример проверяющего модуля

Проверяющий модуль получает данные в следующем формате:

- Строка 1: целые числа n, l, u .
- Строка 2: n целых чисел: w_0, \dots, w_{n-1} .