

تشخیص مولکول‌ها

پتر برای شرکتی کار می‌کند که یک دستگاه برای تشخیص مولکول‌ها ساخته است. وزن هر مولکول یک عدد صحیح مثبت است. دستگاه یک بازه‌ی تشخیص $[l, u]$ دارد، که l و u اعداد صحیح مثبت هستند. دستگاه می‌تواند یک مجموعه از مولکول‌ها را تشخیص دهد، اگر و تنها اگر این مجموعه شامل زیرمجموعه‌ای از مولکول‌ها باشد که مجموع وزن آن‌ها در بازه‌ی تشخیص دستگاه قرار داشته باشد.

به عبارت دیگر، n مولکول با وزن‌های w_0, \dots, w_{n-1} را در نظر بگیرید. عملیات تشخیص موفقیت‌آمیز است اگر مجموعه‌ای از اندیس‌های متمایز $I = \{i_1, \dots, i_m\}$ وجود داشته باشد طوری که $l \leq w_{i_1} + \dots + w_{i_m} \leq u$.

با توجه به ویژگی‌های دستگاه، می‌دانیم که فاصله‌ی بین l و u قطعاً بزرگ‌تر یا مساوی فاصله‌ی بین وزن سنگین‌ترین و سبک‌ترین مولکول است. به عبارت دیگر، $u - l \geq w_{max} - w_{min}$ ، که در آن $w_{max} = \max(w_0, \dots, w_{n-1})$ و $w_{min} = \min(w_0, \dots, w_{n-1})$.

شما باید برنامه‌ای بنویسید که زیرمجموعه‌ای از مولکول‌ها را که مجموع وزن آن‌ها در بازه‌ی تشخیص قرار دارد بیابد، یا این که تعیین کند چنین زیرمجموعه‌ای وجود ندارد.

جزئیات پیاده‌سازی

شما باید تابع زیر را پیاده‌سازی کنید:

```
int[] solve(int l, int u, int[] w) •
```

- l و u : ابتدا و انتهای بازه‌ی تشخیص هستند،
- w : وزن مولکول‌ها است،
- اگر زیرمجموعه‌ی موردنظر وجود داشته باشد، تابع باید یک آرایه از اندیس‌های مولکول‌هایی را برگرداند که چنین مجموعه‌ای را می‌سازد. اگر بیش از یک جواب درست وجود داشته باشد، می‌توانید یکی از آن‌ها را به دل‌خواه برگردانید.
- اگر زیرمجموعه‌ی موردنظر وجود نداشته باشد، تابع باید یک آرایه‌ی خالی برگرداند.

برای زبان C، تعریف تابع کمی متفاوت است:

```
int solve(int l, int u, int[] w, int n, int[] result) •
```

- n : تعداد اعضای w (یعنی تعداد مولکول‌ها) است،
- بقیه‌ی پارامترها همانند بالا است.
- به جای برگرداندن یک آرایه از m اندیس (همانند بالا)، تابع باید اندیس‌ها را در اولین m خانه‌ی آرایه‌ی $result$ بنویسد و سپس m را برگرداند.

• اگر زیرمجموعه‌ی موردنظر وجود نداشت، تابع نباید چیزی را در آرایه‌ی result بنویسد و باید مقدار ۰ را برگرداند.

برنامه‌ی شما می‌تواند اندیس‌ها را به هر ترتیب دل‌خواه در آرایه‌ی بازگشتی (یا در آرایه‌ی result در C) قرار دهد. برای جزئیات پیاده‌سازی در زبان برنامه‌سازی موردنظر خود، لطفاً از فایل‌های قالب داده‌شده استفاده کنید.

مثال‌ها

مثال ۱

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

در این مثال چهار مولکول با وزن‌های ۶، ۸، ۸ و ۷ داده شده است. دستگاه می‌تواند هر زیرمجموعه از مولکول‌ها را که مجموع وزن آن‌ها در بازه‌ی بسته‌ی ۱۵ تا ۱۷ قرار دارد تشخیص دهد. دقت کنید که $۸ - ۶ \geq ۱۵ - ۱۷$. به طور مثال، مجموع وزن مولکول‌های ۱ و ۳ برابر است با $۱۵ = ۸ + ۷ = w_1 + w_3$ ، بنابراین تابع می‌تواند آرایه‌ی $[1, 3]$ را برگرداند. جواب‌های ممکن دیگر عبارت‌اند از $[1, 2]$ ($۱۶ = ۸ + ۸ = w_1 + w_2$) و $[2, 3]$ ($۱۵ = ۸ + ۷ = w_2 + w_3$).

مثال ۲

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

در این مثال چهار مولکول با وزن‌های ۵، ۵، ۶ و ۶ داده شده است و ما به دنبال زیرمجموعه‌ای از آن‌ها با مجموع وزن بین ۱۴ و ۱۵ هستیم. مجدداً دقت کنید که $۶ - ۵ \geq ۱۴ - ۱۵$. هیچ زیرمجموعه‌ای از مولکول‌ها با مجموع وزن ۱۴ یا ۱۵ وجود ندارد، بنابراین تابع باید یک آرایه‌ی تهی برگرداند.

مثال ۳

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

در این مثال چهار مولکول با وزن‌های ۱۵، ۱۷، ۱۶ و ۱۸ داده شده است و ما به دنبال زیرمجموعه‌ای از آن‌ها با مجموع وزن بین ۱۰ و ۲۰ هستیم. مجدداً، دقت کنید که $۱۸ - ۱۵ \geq ۱۰ - ۲۰$. هر زیرمجموعه‌ی یک‌عضوی مجموع وزنی بین ۱۰ و ۲۰ دارد، بنابراین جواب‌های درست عبارت‌اند از $[0]$ ، $[1]$ ، $[2]$ و $[3]$.

زیرمسئله‌ها

۱. (۹ امتیاز): $۱ \leq n \leq ۱۰۰$ ، $۱ \leq w_i \leq ۱۰۰$ ، $۱ \leq u, l \leq ۱۰۰۰$ و همه‌ی w_i ها برابرند.

۲. (۱۰ امتیاز): $۱ \leq n \leq ۱۰۰$ ، $۱ \leq w_i, u, l \leq ۱۰۰۰$ و $\max(w_0, \dots, w_{n-1}) - \min(w_0, \dots, w_{n-1}) \leq ۱$

۳. (۱۲ امتیاز): $۱ \leq n \leq ۱۰۰$ و $۱ \leq w_i, u, l \leq ۱۰۰۰$

۴. (۱۵ امتیاز): $۱ \leq n \leq ۱۰۰۰۰$ و $۱ \leq w_i, u, l \leq ۱۰۰۰۰$

۵. (۲۳ امتیاز): $۱ \leq n \leq ۱۰۰۰۰$ و $۱ \leq w_i, u, l \leq ۵۰۰۰۰۰$

۶. (۳۱ امتیاز): $۱ \leq n \leq ۲۰۰۰۰۰$ و $۱ \leq w_i, u, l < ۲^{۳۱}$

ارزیاب نمونه

ارزیاب نمونه، ورودی را در قالب زیر می خواند:

- خط ۱: اعداد صحیح n ، l و u .
- خط ۲: n عدد صحیح: w_0, \dots, w_{n-1} .