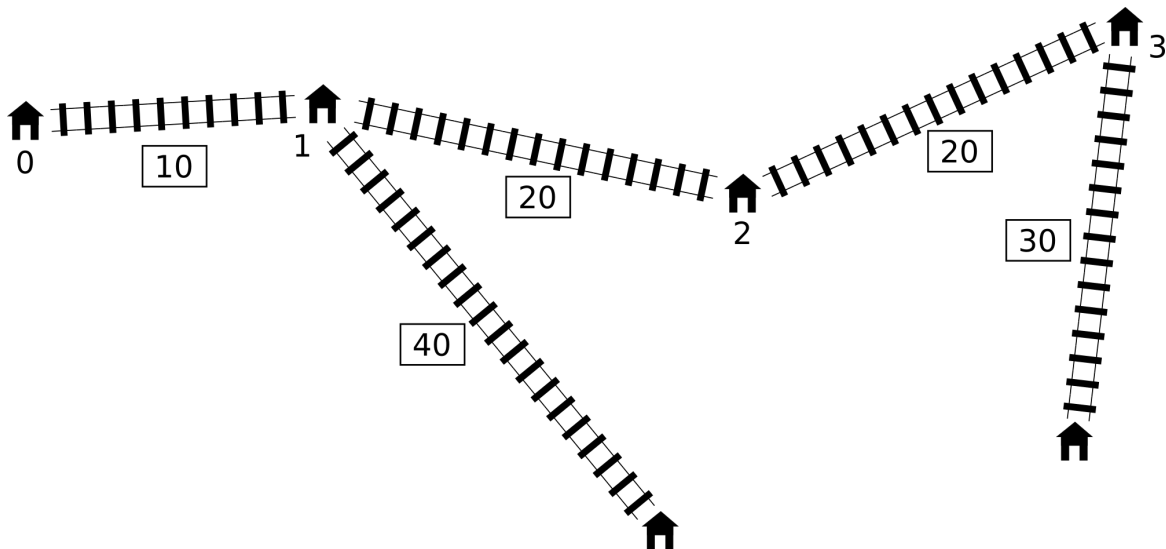


میان‌بر

پاول یک راه‌آهن اسباب‌بازی دارد که خیلی ساده است: یک خط اصلی شامل n ایستگاه، که در طول خط به ترتیب از صفر تا $n - 1$ شماره‌گذاری شده‌اند. فاصله‌ی بین ایستگاه‌های i و $i + 1$ برابر l_i سانتی‌متر است ($0 \leq i < n - 1$).

علاوه‌بر خط اصلی، ممکن است چند خط جانبی هم وجود داشته باشد. هر خط جانبی یک خط آهن بین یک ایستگاه در خط اصلی و یک ایستگاه جدید است که روی خط اصلی قرار ندارد (ایستگاه‌های جدید شماره‌گذاری نشده‌اند). از هر ایستگاه خط اصلی، حداکثر یک خط جانبی می‌تواند آغاز شود. طول خط جانبی که از ایستگاه i شروع می‌شود d_i سانتی‌متر است. اگر هیچ خط جانبی از ایستگاه i آغاز نشود $d_i = 0$ خواهد بود.



اکنون، پاول در نظر دارد یک میان‌بر بسازد: یک خط سریع‌السیر بین دو ایستگاه مختلف خط اصلی (که می‌توانند همسایه هم باشند). طول خط سریع‌السیر دقیقاً c سانتی‌متر است، مستقل از آن که کدام دو ایستگاه را به هم وصل می‌کند.

هر قسمت از راه‌آهن، شامل خط سریع‌السیر جدید، دو طرفه است. فاصله‌ی بین دو ایستگاه، کم‌ترین طول مسیری است که روی راه‌آهن از یکی به دیگری می‌رود. قطر کل شبکه‌ی راه‌آهن، بیشینه فاصله‌ی میان همه‌ی جفت ایستگاه‌ها است؛ به عبارت دیگر، کوچک‌ترین عدد t است طوری که فاصله‌ی بین هر جفت ایستگاه حداکثر t باشد.

پاول می‌خواهد خط سریع‌السیر را طوری بسازد که قطر شبکه‌ی حاصل کمینه باشد.

جزئیات پیاده‌سازی

شما باید تابع زیر را پیاده‌سازی کنید:

```
int64 find_shortcut(int n, int[] l, int[] d, int c)
```

- n : تعداد ایستگاه‌های خط اصلی
 - l : فاصله‌ی بین ایستگاه‌های خط اصلی (آرایه به طول $n - 1$)
 - d : طول خط‌های جانبی (آرایه به طول n)
 - c : طول خط سریع‌السیر جدید
 - تابع باید کمترین قطر ممکن شبکه‌ی راه‌آهن را بعد از افزودن خط سریع‌السیر برگرداند.
- لطفاً از فایل‌های قالب ارائه‌شده برای جزئیات پیاده‌سازی در زبان برنامه‌نویسی خودتان استفاده کنید.

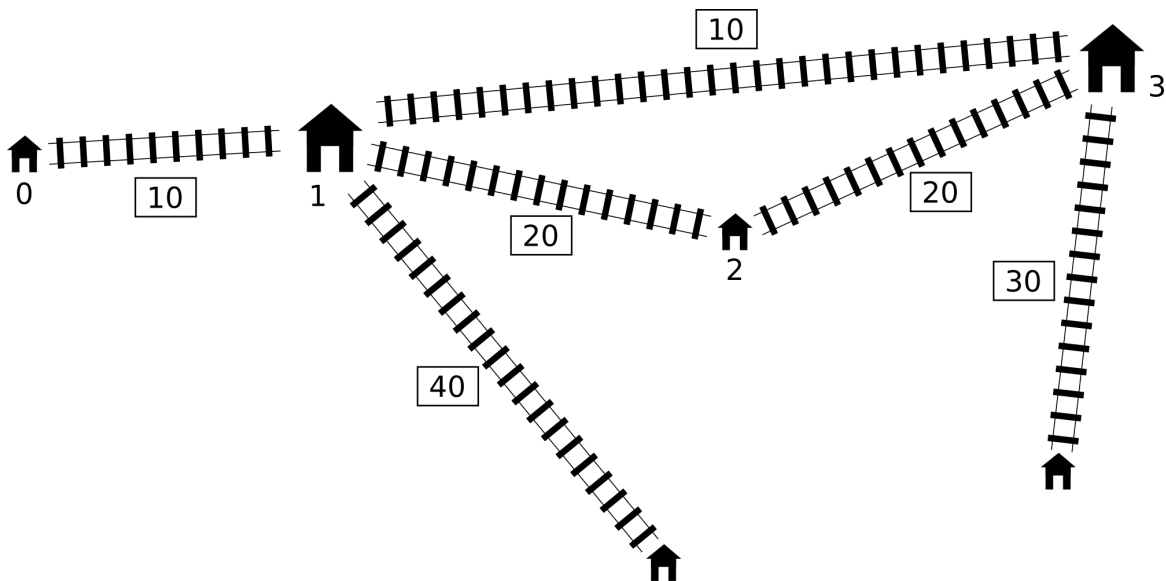
مثال‌ها

مثال ۱

برای شبکه‌ی راه‌آهنی که در بالا نمایش داده‌شد، ارزیاب فراخوانی تابع را به صورت زیر انجام می‌دهد:

```
find_shortcut(4, [10, 20, 20],
              [0, 40, 0, 30], 10)
```

راه‌حل بهینه، ساختن خط سریع‌السیر بین ایستگاه‌های ۱ و ۳ است، به صورتی که در شکل زیر نمایش داده شده.



قطر شبکه‌ی راه‌آهن جدید ۸۰ سانتی‌متر است، بنابراین تابع باید عدد ۸۰ را برگرداند.

مثال ۲

ارزیاب، فراخوانی تابع زیر را انجام می‌دهد:

```
find_shortcut(9, [10, 10, 10, 10, 10, 10, 10, 10],
              [20, 0, 30, 0, 0, 40, 0, 40, 0], 30)
```

راه حل بهینه، متصل کردن ایستگاه‌های ۲ و ۷ است؛ که در این حالت قطر ۱۱۰ خواهد بود.

مثال ۳

ارزیاب، فراخوانی تابع زیر را انجام می‌دهد:

```
find_shortcut(4, [2, 2, 2],
              [1, 10, 10, 1], 1)
```

راه حل بهینه اتصال ایستگاه‌های ۱ و ۲ است که قطر را به ۲۱ کاهش می‌دهد.

مثال ۴

ارزیاب، فراخوانی تابع زیر را انجام می‌دهد:

```
find_shortcut(3, [1, 1],
              [1, 1, 1], 3)
```

متصل کردن هر جفت ایستگاه با خط سریع‌السیر به طول ۳ قطر اولیه‌ی شبکه‌ی راه‌آهن را که ۴ است بهبود نمی‌دهد.

زیرمسئله‌ها

در همه‌ی زیرمسئله‌ها $2 \leq n \leq 1000000$ ، $1 \leq l_i \leq 10^9$ ، $0 \leq d_i \leq 10^9$ و $1 \leq c \leq 10^9$ است.

۱. (۹ امتیاز) $2 \leq n \leq 10$

۲. (۱۴ امتیاز) $2 \leq n \leq 100$

۳. (۸ امتیاز) $2 \leq n \leq 250$

۴. (۷ امتیاز) $2 \leq n \leq 500$

۵. (۳۳ امتیاز) $2 \leq n \leq 3000$

۶. (۲۲ امتیاز) $2 \leq n \leq 100000$

۷. (۴ امتیاز) $2 \leq n \leq 300000$

۸. (۳ امتیاز) $2 \leq n \leq 1000000$

ارزیاب نمونه

ارزیاب نمونه ورودی را به صورت زیر می خوانند:

- سطر ۱: اعداد صحیح n و c
- سطر ۲: اعداد صحیح l_0, l_1, \dots, l_{n-2}
- سطر ۳: اعداد صحیح d_0, d_1, \dots, d_{n-1}