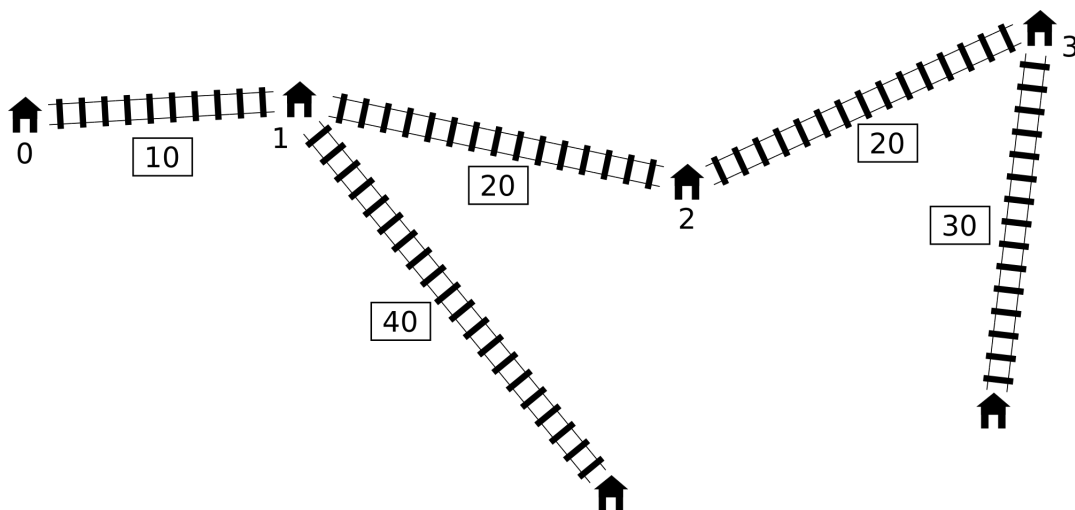


Đường tắt (Shortcut)

Pavel có bộ đồ chơi đường sắt. Nó rất đơn giản. Có một đường chính bao gồm n trạm. Các trạm được đánh số từ 0 đến $n - 1$ theo thứ tự dọc theo đường chính. Khoảng cách giữa các trạm i và $i + 1$ là l_i cm ($0 \leq i < n - 1$).

Ngoài đường chính có thể có một số đường nhánh. Mỗi đường nhánh là một đường sắt giữa một trạm trên đường chính và một trạm mới không nằm trên đường chính. (Các trạm mới không được đánh số). Có nhiều nhất một đường nhánh xuất phát từ mỗi trạm của đường chính. Độ dài của đường nhánh bắt đầu từ trạm i là d_i cm.

Chúng ta dùng $d_i = 0$ để biểu thị rằng không có đường nhánh xuất phát từ trạm i .



Pavel đang lên kế hoạch để xây dựng một đường tắt: một đường cao tốc giữa hai trạm khác nhau (có thể kế nhau) của **đường chính**. Đường cao tốc sẽ có độ dài đúng bằng c cm, bất kể hai trạm nào được kết nối.

Mỗi đoạn của đường sắt, bao gồm cả đường cao tốc mới, có thể đi theo cả hai hướng. *Khoảng cách* giữa hai trạm là độ dài nhỏ nhất của tuyến đường đi từ trạm này đến trạm kia dọc theo đường sắt. *Đường kính* của mạng lưới đường sắt là khoảng cách lớn nhất giữa tất cả các cặp trạm. Nói cách khác, đây là số nhỏ nhất t mà khoảng cách giữa mỗi cặp trạm không vượt quá t .

Pavel muốn xây dựng đường cao tốc mà đường kính của mạng thu được là nhỏ nhất.

Chi tiết cài đặt

Bạn cần cài đặt hàm sau

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

- **n**: số lượng trạm trên đường chính,
- **l**: độ dài giữa các trạm trên đường chính (mảng độ dài $n - 1$),
- **d**: độ dài của các đường nhánh (mảng độ dài n),
- **c**: độ dài của đường cao tốc mới.
- hàm trả về đường kính nhỏ nhất có thể của mạng đường sắt sau khi thêm đường cao tốc.

Hãy sử dụng các file nền được cung cấp cho việc cài đặt chi tiết trên ngôn ngữ lập trình của bạn.

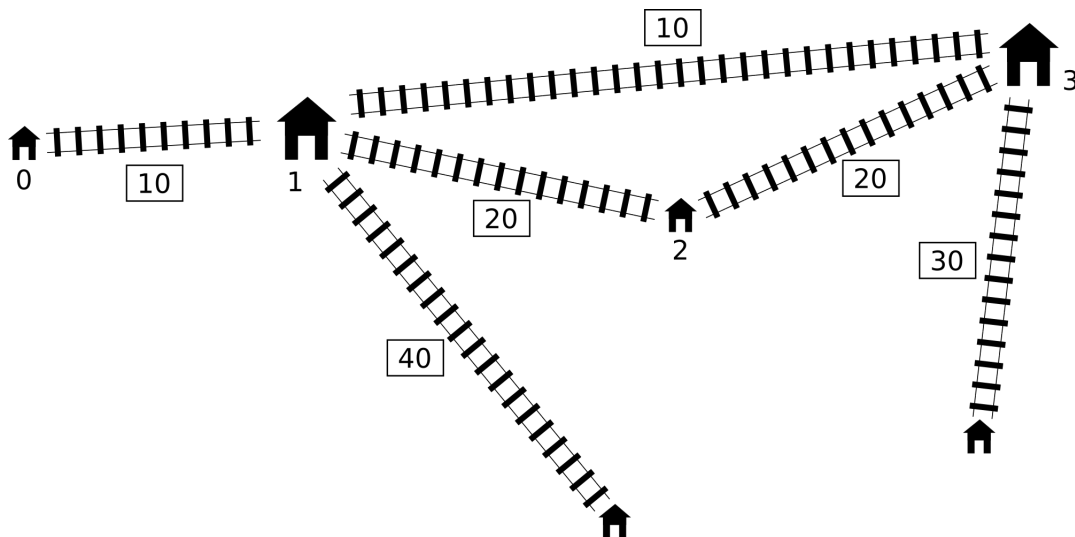
Ví dụ

Ví dụ 1

Với mạng đường sắt ở hình vẽ bên trên, grader sẽ thực hiện lệnh gọi hàm dưới đây:

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

Phương án tối ưu là xây dựng đường cao tốc giữa trạm 1 và 3 như hình dưới đây.



Đường kính của mạng lưới đường sắt mới là 80 cm, do đó hàm sẽ trả về 80.

Ví dụ 2

grader sẽ thực hiện lệnh gọi hàm dưới đây:

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

Phương án tối ưu là nối trạm 2 và 7, trong trường hợp này đường kính là 110.

Ví dụ 3

grader gọi hàm dưới đây:

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

Phương án tối ưu là nối trạm 1 và 2, giảm đường kính về 21.

Ví dụ 4

grader gọi hàm dưới đây:

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

Kết nối hai trạm bất kỳ với đường cao tốc độ dài 3 không cải thiện đường kính ban đầu của mạng lưới đường sắt là 4.

Subtasks

Với tất cả các Subtask $2 \leq n \leq 1000000$, $1 \leq l_i \leq 10^9$, $0 \leq d_i \leq 10^9$, $1 \leq c \leq 10^9$.

1. (9 điểm) $2 \leq n \leq 10$,
2. (14 điểm) $2 \leq n \leq 100$,
3. (8 điểm) $2 \leq n \leq 250$,
4. (7 điểm) $2 \leq n \leq 500$,
5. (33 điểm) $2 \leq n \leq 3000$,
6. (22 điểm) $2 \leq n \leq 100000$,
7. (4 điểm) $2 \leq n \leq 300000$,
8. (3 điểm) $2 \leq n \leq 1000000$.

Sample grader

Chương trình grader mẫu đọc dữ liệu đầu vào theo định dạng sau:

- dòng 1: các số nguyên n và c ,
- dòng 2: các số nguyên l_0, l_1, \dots, l_{n-2} ,
- dòng 3: các số nguyên d_0, d_1, \dots, d_{n-1} .