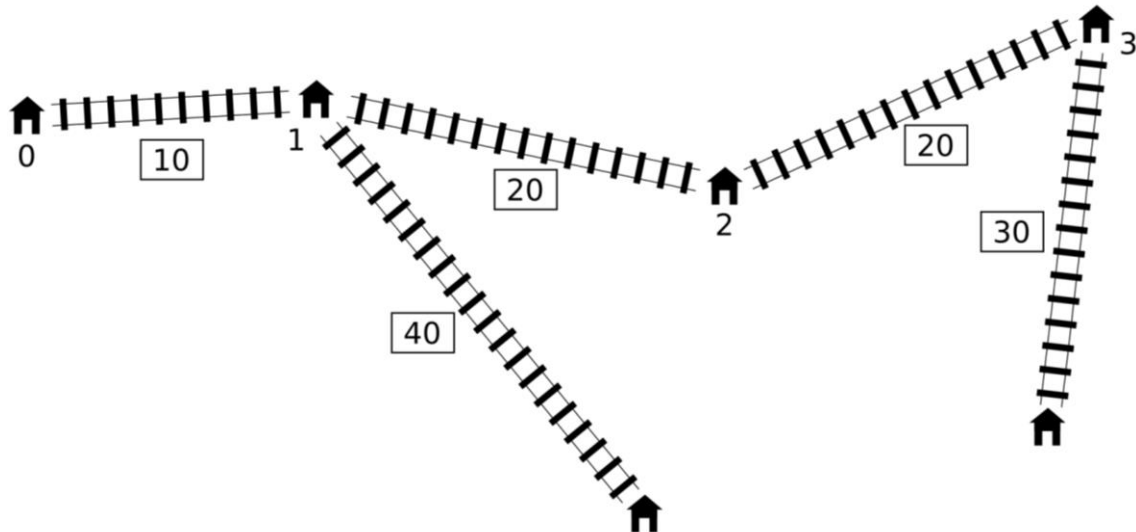


Shortcut

לפאבל (Pavel) יש משחק פסי רכבת. פסי הרכבת מאוד פשוטים. יש קו ראשי אחד (main line) המורכב מ- n תחנות. התחנות ממוספרות מ-0 עד $n - 1$ לפי הסדר שלהן לאורך הקו. המרחק בין התחנות i ו- $i + 1$ הוא l_i סנטימטרים ($0 \leq i < n - 1$).

מלבד הקו הראשי עשויים להיות קווים משניים (secondary lines). כל קו משני מחבר בין תחנה הנמצאת על הקו הראשי ותחנה חדשה שאינה נמצאת על הקו הראשי. (התחנות החדשות האלה אינן ממוספרות). מכל תחנה בקו הראשי יכול לצאת קו משני אחד לכל היותר. האורך של קו משני המתחיל בתחנה i הוא d_i סנטימטרים. נשתמש ב- $d_i = 0$ כדי לסמן שלא קיים קו משני שמתחיל בתחנה i .



פאבל מתכנן לבנות קיצור דרך אחד: קו אקספרס בין שתי תחנות שונות על הקו הראשי (ייתכן שהן שכנות). קו האקספרס יהיה באורך c סנטימטרים בדיוק, ללא תלות בשתי התחנות שהוא מחבר.

כל מקטע של פסי הרכבת, כולל קו האקספרס החדש, הוא דו-כיווני. המרחק בין שתי תחנות הוא האורך הקטן ביותר של מסלול שעובר מתחנה אחת לשנייה לאורך פסי הרכבת. הקוטר של כל רשת פסי הרכבת הוא המרחק המקסימלי בין כל זוגות התחנות. במילים אחרות, זהו המספר t הקטן ביותר, כך שהמרחק בין כל זוג תחנות הוא לכל היותר t .

פאבל רוצה לבנות את קו האקספרס כך שהקוטר של רשת פסי הרכבת יהיה מינימלי.

פרטי מימוש

עליכם לממש את הפונקציה הבאה:

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

- n : מספר התחנות על הקו הראשי.
- l : המרחקים בין התחנות על הקו הראשי (מערך בגודל $n - 1$).
- d : אורכי הקווים המשניים (מערך בגודל n).
- c : אורך קו האקספרס החדש.
- הפונקציה צריכה להחזיר את הקוטר המינימלי האפשרי של רשת פסי הרכבת אחרי הוספת קו האקספרס.

אנא השתמשו בקבצי ה-`template` לפרטי המימוש בשפת התכנות שלכם.

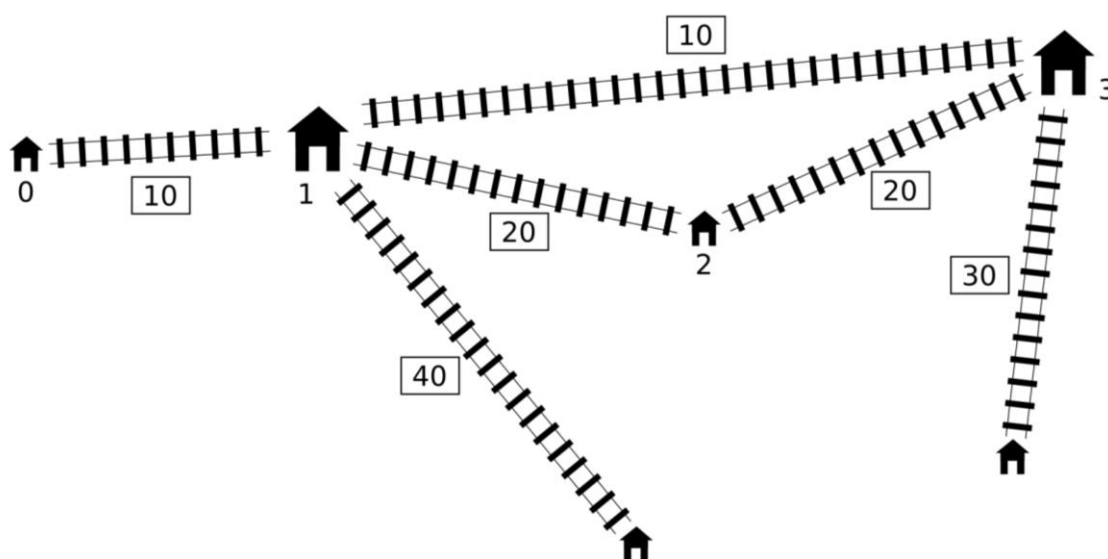
דוגמאות

דוגמה 1

לרשת פסי הרכבת באיור לעיל, הגריידר יבצע את הקריאה הבאה :

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

הפתרון האופטימלי הוא לבנות את קו האקספרס בין תחנות 1 ו-3, כמתואר באיור הבא.



הקוטר של רשת הפסים החדשה הוא 80 סנטימטר, ולכן הפונקציה צריכה להחזיר 80.

דוגמה 2

הגריידר מבצע את הקריאה הבאה :

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

הפתרון האופטימלי הוא לחבר את תחנות 2 ו-7. במקרה זה הקוטר הוא 110.

דוגמה 3

הגריידר מבצע את הקריאה הבאה :

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

הפתרון האופטימלי הוא לחבר את תחנות 1 ו-2, וכך להקטין את הקוטר ל-21.

דוגמה 4

הגריידר מבצע את הקריאה הבאה :

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

חיבור של כל שתי תחנות בקו אקפרס באורך 3 לא משפר את הקוטר ההתחלתי של רשת פסי הרכבת, שהוא 4.

תת משימות

בכל תת המשימות מתקיים $2 \leq n \leq 1\,000\,000$, $1 \leq l_i \leq 10^9$, $0 \leq d_i \leq 10^9$, $1 \leq c \leq 10^9$.

1. 9 נקודות) : $2 \leq n \leq 10$.
2. 14 נקודות) : $2 \leq n \leq 100$.
3. 8 נקודות) : $2 \leq n \leq 250$.
4. 7 נקודות) : $2 \leq n \leq 500$.
5. 33 נקודות) : $2 \leq n \leq 3000$.
6. 22 נקודות) : $2 \leq n \leq 100\,000$.
7. 4 נקודות) : $2 \leq n \leq 300\,000$.
8. 3 נקודות) : $2 \leq n \leq 1\,000\,000$.

גריידר לדוגמה

הגריידר קורא קלט בפורמט הבא :

- שורה ראשונה : המספר השלם n ואחריו c .
- שורה שנייה : המספרים השלמים l_0, l_1, \dots, l_{n-2} .
- שורה שלישית : המספרים השלמים d_0, d_1, \dots, d_{n-1} .