

Състезание

Във връзка с МОИ, град Патая ще бъде домакин и на друго състезание: *Международната олимпиада по надбягване* (МОН). Затова домакините трябва да намерят най-доброто възможно трасе за състезанието.

В областта Патая-Чонбури, има N града, свързани с $N - 1$ магистрала. Всяка магистрала е двупосочна, свързва два различни града и дължината ѝ е цяло число километри. За всеки два града на областта съществува *единствен път*, свързващ тези два града. Т. е. съществува единствен начин да се стигне от единия град до другия по редица от магистрала, без да се влиза в някой от градовете повече от един път.

Правилата на МОН изискват трасето на състезанието да е с дължина *точно* K километра и да завършва в град, който е различен от началния. Нито една от магистралите (а значи и никой от градовете) не може да бъде използвана два пъти в трасето, за да се избягнат сблъсквания. За да не се пречи, пък, твърде много на трафика, трасето на състезанието трябва да съдържа колкото може по-малко магистрала.

Задача

Напишете функция **best_path(N,K,H,L)**, със следните параметри:

- N - броят на градовете, номерирани от 0 до $N - 1$.
- K - исканата дължина на трасето.
- H - двумерен масив, задаващ магистралите. За $0 \leq i < N - 1$, магистралата с номер i свързва градовете $H[i][0]$ и $H[i][1]$.
- L - едномерен масив, задаващ дължината на магистралите. За $0 \leq i < N - 1$, дължината на магистралата с номер i е $L[i]$.

Всички стойности на масива H са между 0 и $N - 1$, включително, а зададената от тях мрежа от магистрала отговаря на условията описани по-горе. Стойностите на масива L са цели числа между 0 и $1\,000\,000$ включително.

Функцията трябва да връща *минималния брой магистрала*, които съставят допустимо трасе с дължина K . Ако такова трасе не съществува, функцията трябва да върне -1 .

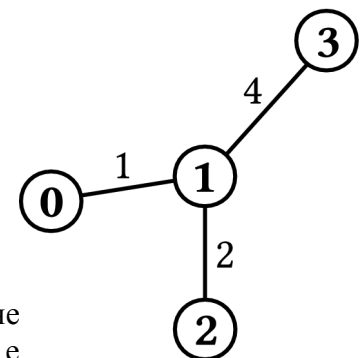
Примери

Пример 1

За мрежата от Фиг. 1, $N=4$, $K=3$,

	0	1		1
$H=$	1	2	$L=$	2
	1	3		4

Трасето на състезанието може да започне в града с номер 0 , да мине през града с номер 1 и да завърши в града с номер 2 . Дължината му е точно $1\text{ km} + 2\text{ km} = 3\text{ km}$ и е съставено от две магистрала. Това е най-доброто възможно трасе, затова **best_path(N,K,H,L)** трябва да върне стойност 2 .



Фигура 1.

Подзадача 3 (22 т.)

- $1 \leq N \leq 200\,000$
- $1 \leq K \leq 100$

Подзадача 4 (57 т.)

- $1 \leq N \leq 200\,000$
- $1 \leq K \leq 1\,000\,000$

Детайли на реализацията

Ограничения

- По време: 3 сек.
- По памет: 256 МВ.

Забележка: Няма ограничение на размера на стека в явен вид. Паметта отделена за стек е част от общата достъпна памет.

Интерфейс (API)

- Папка на задачата: `race/`
- Файл с решение на състезателя: `race.c` или `race.cpp` или `race.pas`
- Интерфейсен файл на състезателя: `race.h` или `race.pas`
- Интерфейсен файл на грейдъра: `race.h` или `racelib.pas`
- Примерен грейдър: `grader.c` или `grader.cpp` или `grader.pas`
- Примерни входни файлове на грейдъра: `grader.in.1`, `grader.in.2`, ...

Забележка: Примерният грейдър чете входни файлове със следния формат:

- Ред 1: N и K .
- Редове от 2 до N : данни за магистралите, за $0 \leq i < N - 1$, редът с номер $i + 2$ съдържа $H[i][0]$, $H[i][1]$ и $L[i]$, разделени с по един интервал.
- Ред $N + 1$: очакваното решение.
- Файлове с очакваните резултати за примерния грейдър: `grader.expect.1`, `grader.expect.2`, ... Всеки от тези файлове трябва да съдържа точно текста “**Correct.**”.