

## Гонки

Наряду с IOI в Паттайе проходят *международные олимпийские гонки* (МОГ) 2011. Принимающей стороне требуется найти наиболее подходящую трассу для гонок.

В регионе Паттайя-Чонбури находятся  $N$  городов, соединённых сетью из  $(N-1)$  магистралей. Каждая магистраль — двусторонняя, соединяет два различных города, и для неё известна длина в километрах — целое число. Известно, что между каждой парой городов существует *ровно один* возможный путь, соединяющий эти города. Таким образом, для любой пары городов существует ровно одна последовательность различных магистралей, по которой можно проехать из одного города в другой, не посещая никакой город дважды.

По требованиям организаторов МОГ трасса должна являться путём суммарной длины *ровно*  $K$  километров, начинающимся и заканчивающимся в различных городах. Естественно, никакая магистраль и, поэтому, никакой город не могут быть использованы дважды при выборе трассы, иначе возможны столкновения. Чтобы минимизировать влияние гонок на трафик движения в регионе, необходимо выбрать для трассы путь из наименьшего возможного количества магистралей.

### Задание

Написать процедуру `best_path(N,K,H,L)`, которой передаются следующие параметры:

- $N$  — количество городов. Города нумеруются от  $0$  до  $(N-1)$ .
- $K$  — требуемая длина трассы.
- $H$  — двумерный массив, описывающий магистрали. Для  $0 \leq i < N-1$  магистраль с номером  $i$  соединяет города с номерами  $H[i][0]$  и  $H[i][1]$ .
- $L$  — одномерный массив, содержащий длины магистралей. Для  $0 \leq i < N-1$  длина магистрали с номером  $i$  равна  $L[i]$ .

Гарантируется, что все значения в массиве  $H$  лежат в пределах от  $0$  до  $(N-1)$  включительно, и описанные в этом массиве магистрали соединяют все города, как описано выше. Также гарантируется, что все значения в массиве  $L$  — целые числа в пределах от  $0$  до  $1\,000\,000$  включительно.

Ваша процедура должна возвращать *минимальное возможное количество магистралей* на допустимой трассе, имеющей длину, равную  $K$ . Если такой трассы не существует, ваша процедура должна вернуть значение  $-1$ .

## Примеры

### Пример 1

Рассмотрим пример, представленный на рис. 1, где  $N=4$ ,  $K=3$ ,

0 1	1
<b>H=</b> 1 2	<b>L=</b> 2
1 3	4

Трасса может начинаться в городе с номером 0, проходить через город с номером 1 и закончиться в городе с номером 2. Длина трассы будет равна  $(1 + 2) = 3$  км, как и требуется, и она состоит из двух магистралей. Это наилучшая возможная трасса; поэтому процедура `best_path(N,K,H,L)` должна вернуть значение 2.

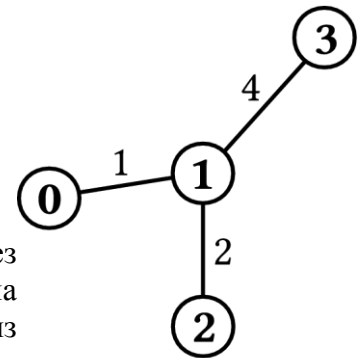


Рис. 1

### Пример 2

Рассмотрим пример, представленный на рис. 2, где  $N=3$ ,  $K=3$ ,

0 1	1
<b>H=</b> 1 2	<b>L=</b> 1

Здесь допустимой трассы не существует. В этом примере процедура `best_path(N,K,H,L)` должна вернуть значение -1.

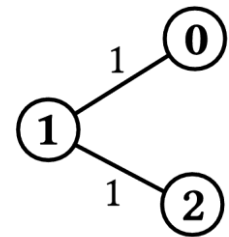


Рис. 2

### Пример 3

Рассмотрим пример, представленный на рис. 3, где  $N=11$ ,  $K=12$ ,

0 1	3
0 2	4
2 3	5
3 4	4
4 5	6
<b>H=</b> 0 6	<b>L=</b> 3
6 7	2
6 8	5
8 9	6
8 10	7

Одна из возможных трасс состоит из 3 магистралей: она идёт из города с номером 6 через города с номерами 0 и 2 в город с номером 3. Другая трасса начинается в городе с номером 10 и идёт через город с номером 8 в город с номером 6. Обе эти трассы имеют длину 12 километров, как и требуется. Вторая из них оптимальна, так как не существует подходящей трассы из одной магистрали. Таким образом, процедура `best_path(N,K,H,L)` должна вернуть значение 2.

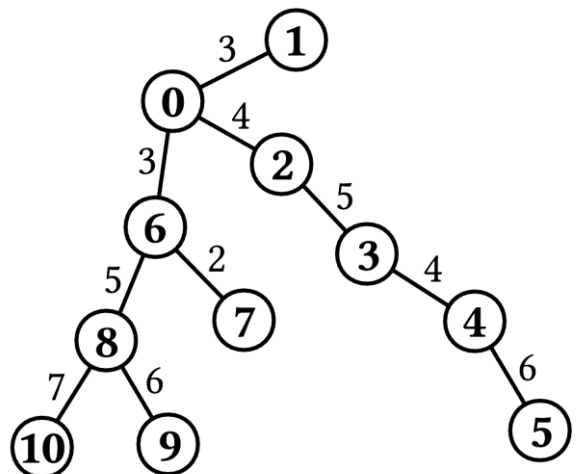


Рис. 3

## Подзадачи

### Подзадача 1 (9 баллов)

- $1 \leq N \leq 100$
- $1 \leq K \leq 100$
- Сеть магистралей представляет собой линию: для  $0 \leq i < N-1$ , магистраль с номером  $i$  соединяет города с номерами  $i$  и  $i+1$ .

### Подзадача 2 (12 баллов)

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

### Подзадача 3 (22 балла)

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

### Подзадача 4 (57 баллов)

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

## Детали реализации

### Ограничения

- Ограничение по времени: 3 секунды
- Ограничение по памяти: 256 МВ  
**Замечание:** Нет отдельного ограничения на размер стека; используемая стеком память входит в общий объём используемой памяти.

### Интерфейс (API)

- Папка для разработки: `race/`
- Участник должен разработать: `race.c` или `race.cpp` или `race.pas`
- Интерфейс участника: `race.h` или `race.pas`
- Интерфейс модуля оценивания: `race.h` или `racelib.pas`
- Предлагаемый модуль оценивания: `grader.c` или `grader.cpp` или `grader.pas`
- Ввод для предлагаемого модуля оценивания: `grader.in.1`, `grader.in.2`, ...

**Замечание:** Предлагаемый модуль оценивания читает входной файл в следующем формате:

- Строка 1:  $N$  и  $K$ .
- Строки от 2 до  $N$ : информация о магистралях; то есть, строка с номером  $(i+2)$  содержит для  $0 \leq i < N-1$  значения  $H[i][0]$ ,  $H[i][1]$  и  $L[i]$ , разделённые одним пробелом.
- Строка  $N+1$ : ожидаемое решение.
- Ожидаемый вывод для предлагаемого модуля оценивания: `grader.expect.1`, `grader.expect.2`, ... В этой задаче каждый из перечисленных файлов должен содержать только текст “**Correct.**”