Подземный город крокодилов

Археолог Бенджама спасается бегством после попытки исследовать таинственный подземный город крокодилов. Город состоит из **N** комнат и **M** двусторонних коридоров, каждый из которых соединяет пару различных комнат. Каждая пара комнат соединена не более, чем одним коридором. Для каждого из коридоров известно время, которое требуется, чтобы по нему пробежать. Только **K** из **N** комнат имеют выходы из города. Бенджама начинает из комнаты с номером **0**. Она хочет добраться до выхода как можно быстрее.

Страж города крокодилов хочет, чтобы Бенджама не смогла убежать. Из своей пещеры страж контролирует секретные двери, которые могут заблокировать *ровно один* коридор. То есть, когда он блокирует какой-то коридор, то предыдущий заблокированный коридор разблокируется.

Бенджама находится в следующей ситуации: каждый раз, когда она пытается покинуть какую-либо комнату, страж может выбрать один из коридоров, связанных с этой комнатой, и заблокировать его. После этого Бенджама выбирает один из незаблокированных коридоров и бежит по нему в другую комнату. Как только Бенджама вбегает в какой-то коридор, страж не может заблокировать этот коридор, пока Бенджама не выбежит из него на другом конце этого коридора. Только после того, как она вбежит в очередную комнату, страж снова может выбрать какой-то связанный с новой комнатой коридор (в том числе и тот, по которому Бенджама только что пробежала) и заблокировать его, и т.д.

Бенджама хотела бы заранее придумать какой-то несложный план побега. А именно, ей требуется набор инструкций, который будет сообщать ей, что делать, когда она попадает в какую-либо комнату. Обозначим эту комнату как **A**. Если из комнаты **A** есть выход из города, то очевидно, что никаких инструкций не требуется — Бенджама может убежать. Иначе, инструкция для комнаты **A** должна иметь одну из следующих форм:

- "Если ты когда-нибудь попадёшь в комнату **A**, убегай по коридору, ведущему в комнату **B**. Однако если этот коридор заблокирован, убегай по коридору, ведущему в комнату **C**".
- "Не беспокойся насчёт комнаты **A** — по плану ты никогда не попадёшь в неё".

Заметим, что в некоторых случаях (например, если по вашему плану Бенджама будет бегать по циклу) страж может помешать Бенджаме добежать до выхода. План называется *хорошим*, если Бенджама гарантированно попадёт в выход за конечное время независимо от действий стража. Для данного хорошего плана обозначим как **T** такое минимальное время, что по прошествии **T** единиц времени Бенджама *гарантированно* сумеет выйти. В этом случае мы говорим, что *для выполнения данного хорошего плана потребуется затратить время* **T**.

Задание

Написать процедуру travel_plan(N,M,R,L,K,P), которой передаются следующие параметры:

- **N** — количество комнат. Комнаты нумеруются числами от **0** до (**N-1**).
- **M** — количество коридоров. Коридоры нумеруются числами от **0** до (**M-1**).
- **R** — двумерный массив целых чисел, который описывает коридоры. Для $\mathbf{0} \leq \mathbf{i} < \mathbf{M}$,

коридор с номером **i** соединяет две различные комнаты с номерами **R[i][0]** и **R[i][1]**. Никакие два коридора не соединяют одну и ту же пару комнат.

- **L** — одномерный массив целых чисел, в котором содержится время, необходимое для пробега по каждому из коридоров. Для $0 \leq i < M$, значение $1 \leq L[i] \leq 1\,000\,000\,000$ есть время, которое требуется Бенджаме, чтобы пробежать по i-му коридору.
- **K** — количество комнат, в которых есть выход из города. Гарантируется, что $1 \leq K \leq N$.
- **P** — одномерный массив из **K** различных целых чисел, соответствующих комнатам с выходами. Для $0 \leq i < K$ значение **P[i]** является номером **i**-й комнаты с выходом. Комната с номером **0** никогда не является комнатой с выходом.

Ваша процедура должна возвращать минимальное возможное время **T**, для которого существует такой хороший план побега, что для выполнения этого плана потребуется затратить время **T**.

Гарантируется, что с каждой комнатой, не содержащей выхода, связаны хотя бы два коридора. Также гарантируется, что в каждом тесте существует хороший план побега, для которого $T \leq 1\,000\,000\,000$.

Примеры

Пример 1

Рассмотрим пример, показанный на рис. 1, где **N=5**, **M=4**, **K=3** и

R=
0 1
0 2
3 2
2 4

L=
2
3
1
4

P=
1
3
4

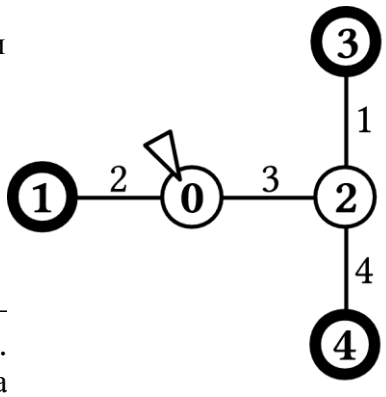

Рис. 1.

Комнаты показаны кружками, а коридоры, соединяющие их — отрезками. Комнаты с выходами обведены жирным контуром. Бенджама начинает движение в комнате с номером 0 (она отмечена треугольником). Оптимальный план побега таков:

- Если ты когда-нибудь попадёшь в комнату с номером 0, убегай по коридору, ведущему в комнату с номером 1. Однако если этот коридор заблокирован, убегай по коридору, ведущему в комнату с номером 2.
- Если ты когда-нибудь попадёшь в комнату с номером 2, убегай по коридору, ведущему в комнату с номером 3. Однако если этот коридор заблокирован, убегай по коридору, ведущему в комнату с номером 4.

В наихудшем случае Бенджама попадёт в комнату с выходом за 7 единиц времени. Таким образом, процедура travel_plan должна возвращать число **7**.

Пример 2

Рассмотрим пример, показанный на рис. 2, где **N=5**, **M=7**, **K=2** и

$$\mathbf{R}=\begin{matrix}0&2\\0&3\\3&2\\2&1\\0&1\\0&4\\3&4\end{matrix}\qquad \mathbf{L}=\begin{matrix}4\\3\\2\\10\\100\\7\\9\end{matrix}\qquad \mathbf{P}=\begin{matrix}1\\3\end{matrix}$$

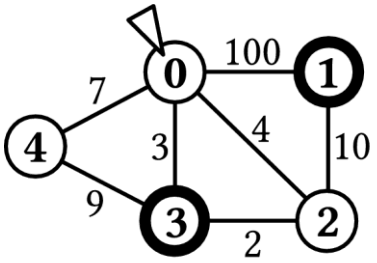

Рис. 2.

Оптимальный план побега описывается следующим образом:

- Если ты когда-нибудь попадёшь в комнату с номером 0, убегай по коридору, ведущему в комнату с номером 3. Однако если этот коридор заблокирован, убегай по коридору, ведущему в комнату с номером 2.
- Если ты когда-нибудь попадёшь в комнату с номером 2, убегай по коридору, ведущему в комнату с номером 3. Однако если этот коридор заблокирован, убегай по коридору, ведущему в комнату с номером 1.
- Не беспокойся насчёт комнаты с номером 4 — по плану ты никогда не попадёшь в неё.

Бенджама попадёт в какую-либо из комнат с выходом не позже, чем по прошествии 14 единиц времени. Поэтому процедура travel_plan должна возвращать значение **14**.

Подзадачи

Подзадача 1 (46 баллов)

- $\mathbf{3 \le N \le 1\,000}$.
- Подземный город представляет собой дерево. То есть, **M** = **N**-1 и для каждой пары комнат с номерами **i** и **j** существует последовательность коридоров, соединяющих комнаты с номерами **i** и **j**.
- Каждая комната с выходом соединена ровно с одной другой комнатой.
- Все остальные комнаты соединены коридорами с тремя или более отличными от неё комнатами.

Подзадача 2 (43 балла)

- $\mathbf{3 \le N \le 1\,000}$.
- $\mathbf{2 \le M \le 100\,000}$.

Подзадача 3 (11 баллов)

- $\mathbf{3 \le N \le 100\,000}$.
- $\mathbf{2 \le M \le 1\,000\,000}$.

Детали реализации

Ограничения

- Ограничение по времени: 2 секунды
- Ограничение по памяти: 256 MB
  Замечание: Нет отдельного ограничения на размер стека; используемая стеком память входит в общий объём используемой памяти.

Интерфейс (API)

- Папка для разработки: crocodile/
- Участник должен разработать: crocodile.c или crocodile.cpp или crocodile.pas
- Интерфейс участника: crocodile.h или crocodile.pas
- Интерфейс модуля оценивания: crocodile.h или crocodilelib.pas
- Предлагаемый модуль оценивания: grader.c или grader.cpp или grader.pas и crocodilelib.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, M и K.
  - Строки от 2 до (**M+1**): Для $\mathbf{0} \leq \mathbf{i} < \mathbf{M}$, строка с номером (i+2) содержит числа R[i][0], R[i][1] и L[i], разделённые пробелом.
  - Строка **M+2**: последовательность K целых чисел P[0], P[1],..., P[K-1], разделённых пробелом.
  - Строка **M+3**: 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.**”