

Задача А. Caves and Tunnels (Пещеры и туннели)

Имя входного файла: caves.in
Имя выходного файла: caves.out
Ограничение по времени: 5 seconds
Ограничение по памяти: 64 megabytes

After landing on Mars surface, scientists found a strange system of caves connected by tunnels. So they began to research it using remote controlled robots. It was found out that there exists exactly one route between every pair of caves. But then scientists faced a particular problem. Sometimes in the caves faint explosions happen. They cause emission of radioactive isotopes and increase radiation level in the cave. Unfortunately robots don't stand radiation well. But for the research purposes they must travel from one cave to another. So scientists placed sensors in every cave to monitor radiation level in the caves. And now every time they move robots they want to know the maximal radiation level the robot will have to face during its relocation. So they asked you to write a program that will solve their problem.

Формат входного файла

The first line of the input contains one integer N ($1 \leq N \leq 100\,000$) — the number of caves. Next $N - 1$ lines describe tunnels. Each of these lines contains a pair of integers a_i, b_i ($1 \leq a_i, b_i \leq N$) specifying the numbers of the caves connected by corresponding tunnel. The next line has an integer Q ($Q \leq 100\,000$) representing the number of queries. The Q queries follow on a single line each. Every query has a form of " $C U V$ ", where C is a single character and can be either 'I' or 'G' representing the type of the query (quotes for clarity only). In the case of an 'I' query radiation level in U th cave ($1 \leq U \leq N$) is incremented by V ($0 \leq V \leq 10\,000$). In the case of a 'G' query your program must output the maximal level of radiation on the way between caves with numbers U and V ($1 \leq U, V \leq N$) after all increases of radiation ('I' queries) specified before current query. It is assumed that initially radiation level is 0 in all caves, and it never decreases with time (because isotopes' half-life time is much larger than the time of observations).

Первая строка входного файла содержит одно целое число N ($1 \leq N \leq 100\,000$) — количество пещер. Следующие $N - 1$ строк описывают туннели. Каждая из этих строк содержит два целых числа — a_i и b_i ($1 \leq a_i, b_i \leq N$), описывающие туннель из пещеры с номером a_i в пещеру с номером b_i . Следующая строка содержит целое число Q ($1 \leq Q \leq 100\,000$), означающее количество запросов. Далее идут Q запросов, по одному на строку. Каждый запрос имеет вид « $C U V$ », где C — символ «I» либо «G», означающие тип запроса (кавычки только для ясности). В случае запроса «I» уровень

радиации в U -й пещере ($1 \leq U \leq N$) увеличивается на V ($0 \leq V \leq 10\,000$). В случае запроса «G» ваша программа должна вывести максимальный уровень радиации на пути между пещерами с номерами U и V ($1 \leq U, V \leq N$) после всех увеличений радиации (запросов «I»), указанных ранее. Предполагается, что изначальный уровень радиации равен 0 во всех пещерах, и он никогда не уменьшается со временем (потому что период полураспада изотопов много больше времени наблюдения).

Формат выходного файла

For every 'G' query output one line containing the maximal radiation level by itself.

Для каждого запроса «G» выведите одну строку, содержащую максимальный уровень радиации.

Пример

caves.in	caves.out
4	1
1 2	0
2 3	1
2 4	3
6	
I 1 1	
G 1 1	
G 3 4	
I 2 3	
G 1 1	
G 3 4	

Задача В. Пещеры и туннели-easy

Задача «Caves and Tunnels (Пещеры и туннели)» с ограничениями $1 \leq N \leq 100$, $1 \leq Q \leq 100$.

Задача С. Верификация моделей

Имя входного файла: model.in
Имя выходного файла: model.out
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Сереза работает над новым проектом по верификации недетерминированных программных моделей. Первая версия программы будет работать с ациклическими программами. Ациклическая программа в виде, пригодном для верификации,

представляется в виде ориентированного ациклического взвешенного графа, в котором есть вершина s , из которой достижимы все остальные.

Результатом верификации ациклической программы является верификационное дерево. *Верификационным деревом* называется ориентированное корневое дерево с корнем в вершине s , по дугам которого можно добраться из вершины s до любой другой вершины. Изучаемой характеристикой верификационного дерева является его характеристика Бабёнок–Копелиовича (ХБК) — количество единиц в двоичной записи суммы весов дуг, из которых оно состоит. Поскольку у ациклической программы может быть несколько верификационных деревьев, Сережу интересует среднее значение ХБК по всем возможным верификационным деревьям.

Формат входного файла

Первая строка входного файла содержит два целых числа n и m — количество вершин и дуг графа, соответственно ($2 \leq n \leq 20$, $1 \leq m \leq 50$). Вершина s имеет номер 1.

Следующие m строк содержат по три целых числа a_i , b_i и c_i — номер вершины, из которой выходит дуга, номер вершины, в которую она входит и веса этой дуги. Веса дуг неотрицательны и не превышают 10^7 . В графе нет параллельных дуг. В графе нет циклов.

Формат выходного файла

Выведите в выходной файл одно вещественное число — среднее количество единиц в двоичной записи веса верификационного дерева. Ответ должен отличаться от правильного не более чем на 10^{-6} .

Примеры

model.in	model.out
4 4 1 2 1 1 3 1 2 4 1 3 4 2	1.5

Задача D. Союзы

Имя входного файла: `alliance.in`
Имя выходного файла: `alliance.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

В Триаметистовом королевстве было n городов и m дорог, соединявших некото-

рые из них друг с другом. Однажды в результате экспериментов придворного мага Д произошло катастрофическое расщепление: во вселенной вместо одного Триаметистового королевства появилось множество его копий, или *отражений*. Более того, в каждом из них каждая дорога стала заколдованной либо способом α , либо способом ω (стоит отметить, что каждое из возможных сочетаний заколдованностей дорог появилось ровно в одном из отражений).

Свойства заколдованностей α и ω таковы, что города a , b и c образуют торговый союз тогда и только тогда, когда есть дороги между a и b , между b и c и между c и a , заколдованные одним и тем же способом.

Формат входного файла

В первой строке входных данных записаны два целых числа n и m — количество городов и дорог Триаметистового королевства. В следующих m строках записаны пары чисел, задающие города, соединённые соответствующими дорогами. $3 \leq n \leq 20\,000$, $1 \leq m \leq 300\,000$. Никакие два города не соединены более чем одной дорогой, никакая дорога не соединяет город сам с собой.

Формат выходного файла

Выведите как можно точнее единственное вещественное число — среднее количество союзов, которое образовалось во всех отражениях Триаметистового королевства.

Пример

alliance.in	alliance.out
3 3 1 2 2 3 3 1	0.25