

Содержание

Обязательные задачи	3
Задача 21А. Пещеры и туннели [1 sec, 256 mb]	3
Задача 21В. Динамический Лес [1 sec, 256 mb]	4
Бонус	5
Задача 21С. Distance Sum [1 sec, 256 mb]	5
Задача 21D. Union [1 sec, 256 mb]	7
Задача 21Е. Inspector is Coming [1 sec, 256 mb]	8

Пример работы с файлами.

Если вы не умеете читать/выводить данные, или открывать файлы, воспользуйтесь примерами. <http://acm.math.spbu.ru/~sk1/algo/sum/>

Пример работы с файлами.

В некоторых задачах большой ввод и вывод. Про ввод-вывод в C++:

http://acm.math.spbu.ru/~sk1/algo/input-output/cpp_common.html

Имеет смысл пользоваться супер быстрым вводом-выводом. Две версии:

http://acm.math.spbu.ru/~sk1/algo/input-output/io_export.cpp.html

http://acm.math.spbu.ru/~sk1/algo/input-output/fread_write_export.cpp.html

Выделение памяти.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) переопределение стандартного аллокатора ускорит вашу программу:

<http://acm.math.spbu.ru/~sk1/algo/memory.cpp.html>

Обязательные задачи

Задача 21А. Пещеры и туннели [1 сек, 256 mb]

После посадки на Марс учёные нашли странную систему пещер, соединённых туннелями. И учёные начали исследовать эту систему, используя управляемых роботов. Было обнаружено, что существует ровно один путь между каждой парой пещер. Но потом учёные обнаружили специфическую проблему. Иногда в пещерах происходят небольшие взрывы. Они вызывают выброс радиоактивных изотопов и увеличивают уровень радиации в пещере. К сожалению, роботы плохо выдерживают радиацию. Но для исследования они должны переместиться из одной пещеры в другую. Учёные поместили в каждую пещеру сенсор для мониторинга уровня радиации. Теперь они каждый раз при движении робота хотят знать максимальный уровень радиации, с которым придётся столкнуться роботу во время его перемещения. Как вы уже догадались, программу, которая это делает, будете писать вы.

Формат входных данных

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 выходных данных

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 21В. Динамический Лес [1 sec, 256 mb]

Вам нужно научиться обрабатывать 3 типа запросов:

1. Добавить ребро в граф (`link`).
2. Удалить ребро из графа (`cut`).
3. По двум вершинам a и b , определить, лежат ли они в одной компоненте связности (`get`).

Изначально граф пустой (содержит N вершин, не содержит ребер). Гарантируется, что в любой момент времени граф является лесом. При добавлении ребра гарантируется, что его сейчас в графе нет. При удалении ребра гарантируется, что оно уже добавлено.

Формат входных данных

Числа N и M ($1 \leq N \leq 10^5 + 1$, $1 \leq M \leq 10^5$) — количество вершин в дереве и, соответственно, запросов. Далее M строк, в каждой строке команда (`link` или `cut`, или `get`) и 2 числа от 1 до N — номера вершин в запросе.

Формат выходных данных

В выходной файл для каждого запроса `get` выведите 0, если не лежат, или 1, если лежат.

Пример

eulertour.in	eulertour.out
3 7 get 1 2 link 1 2 get 1 2 cut 1 2 get 1 2 link 1 2 get 1 2	0101
5 10 link 1 2 link 2 3 link 4 3 cut 3 4 get 1 2 get 1 3 get 1 4 get 2 3 get 2 4 get 3 4	110100

Бонус

Задача 21С. Distance Sum [1 sec, 256 mb]

На некоторой карте обозначены n городов и $n - 1$ дорога, соединяющая эти города таким образом, что полученный граф является деревом. Города занумерованы последовательными целыми числами от 1 до n .

Город 1 является корнем дерева; обозначим для каждого $i > 1$ город, являющийся предком города i , за p_i , а расстояние между городами p_i и i за d_i .

Snuke хочет для каждого $1 \leq k \leq n$ вычислить наименьшую сумму расстояний от некоторого города до городов $1, \dots, k$:

$$\min_{1 \leq v \leq n} \left\{ \sum_{i=1}^k dist(i, v) \right\} \quad (1)$$

Здесь $dist(u, v)$ обозначает расстояние между городами u и v .

Формат входных данных

Первая строка входа содержит одно целое число n ($1 \leq n \leq 2 \cdot 10^5$). Далее идут $n - 1$ строк, i -я из которых содержит два целых числа p_{i+1} и d_{i+1} — номер предка города $i + 1$ и расстояние между городом $i + 1$ и этим предком ($1 \leq p_i \leq n$, $1 \leq d_i \leq 2 \cdot 10^5$, p_i образуют дерево).

Формат выходных данных

Выведите n строк. В i -й из этих строк выведите ответ для $k = i$.

Примеры

tdsum.in	tdsum.out
10	0
4 1	3
1 1	3
3 1	4
3 1	5
5 1	7
6 1	10
6 1	13
8 1	16
4 1	19
15	0
1 3	3
12 5	9
5 2	13
12 1	14
7 5	21
5 1	22
6 1	29
12 1	31
11 1	37
12 4	41
1 1	41
5 5	47
10 4	56
1 2	59

Задача 21D. Union [1 sec, 256 mb]

Дано дерево из n вершин. Нужно обработать запросы вида “количество рёбер на пути от v_i до u_i , вес которых не более k_i ”.

Формат входных данных

На первой строке числе n ($1 \leq n \leq 10^5$). Следующие $n - 1$ строк описывают рёбра дерева. Ребро задаётся парой концов a, b и весом w ($1 \leq a, b \leq n, a \neq b, 1 \leq w \leq 10^6$). Следующая строка содержит число запросов q ($1 \leq q \leq 10^5$). Каждый запрос задаётся тройкой чисел v_i, u_i и k_i ($1 \leq v, u \leq n, 1 \leq k \leq 10^6$).

Формат выходных данных

Для каждого запроса выведите одно число.

Примеры

union.in	union.out
3	1
1 2 1	2
1 3 2	1
3	
1 2 2	
2 3 2	
2 3 1	
4	0
1 2 3	1
2 3 4	3
1 4 6	1
5	0
1 2 2	
4 2 5	
4 3 6	
2 3 5	
2 3 1	

Задача 21E. Inspector is Coming [1 sec, 256 mb]

After the New Great Technical Revolution in 2017 Mankind decided to conquer the entire Universe. In 2030 first spaceships were constructed and several expeditions were organized away from the Solar System. These expeditions were aimed at colonization of new planets.

In 2032 one of the spaceships reached ZetaX and the expedition met the aboriginal population. Colonization of that planet was a hard task. There were desperate struggles for every meter of the territory. As soon as the army conquered a considerable part of the land, they built a base station there to hold the territory and protect it from the enemy. A road to another base station was laid as well. But since any sort of construction during the war is a very risky business, they built exactly one road to an already existing station. There was only one army conquering ZetaX, so base stations appeared successively one by one.

Now the war is over, the army is gone, and N base stations are still on the planet. However, there is still much work to be done. Before ordinary people are allowed to come and settle here, the planet must be checked by a special Imperial inspection. Inspector Mr. Subtle has come to the planet from the Earth. He seems to be very unfriendly. He doesn't want to confirm that the planet is livable until the road system works well enough. Inspector formulated some silly improvements to be made. He tried to make them as hard as possible to get a bribe, but nobody wants to pay him.

The requirements are: it is necessary to improve all roads that are on the way from base station u to base station v and have length from W_{min} to W_{max} . Inspector made Q such requirements.

Find the number of roads it is necessary to improve and Inspector will have to mark ZetaX as livable.

Формат входных данных

The first line of the input file contains N ($2 \leq N \leq 100\,000$) — the number of base stations. Each of the next $N - 1$ lines contains three integers a_i , b_i and w_i ($1 \leq a_i, b_i \leq N$, $1 \leq w_i \leq 1\,000\,000$) — the descriptions of the roads in arbitrary order. The next line contains an integer Q ($0 \leq Q \leq 100\,000$) — the number of requirements. Each of the next Q lines contains four integers u , v , W_{min} , and W_{max} ($1 \leq u, v \leq N$, $1 \leq W_{min} \leq W_{max} \leq 1\,000\,000$) — the requirements themselves.

Формат выходных данных

One integer — the number of roads it is needed to improve.

Примеры

roads.in	roads.out
3 1 2 20 2 3 10 2 1 3 5 15 1 3 15 25	2
4 1 2 10 1 3 20 3 4 30 1 4 2 11 29	1