

Задача А. Мосты: последняя битва

Имя входного файла: bridges3.in
Имя выходного файла: bridges3.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

А у вашей дипломной работы есть практическое применение?

Популярный вопрос

Легенда

Мальчик Серёжа уже почти закончил работу над дипломной работой. Тема диплома с декабря не изменилась, это всё ещё «Dynamic 2-Edge-Connectivity Problem». Алгоритм уже придуман, протестирован, доказана оценка $O(K \log K)$... Осталось только написать главу про «практическое применение».

Ну какое может быть практическое применение у задачи «Dynamic 2-Edge-Connectivity Problem»? Вопрос непростой. Он оказался чуть ли не сложнее исходной задачи. Но диплом нужно скоро защитить, так что срочно нужно что-то делать.

Итак, первое практическое применение: можно предложить на соревнование задачу на эту тему!

Задача

Дан неориентированный граф из не более чем 10^5 вершин. Изначально граф не содержит рёбер. Вам нужно обрабатывать запросы вида ADD x y и DEL x y — добавить или удалить ребро из x в y .

В каждый момент времени нужно знать, **сколько в графе мостов**.

Кратных рёбер и петель ни в какой момент времени нет.

Если поступает команда «удалить ребро», оно в графе точно присутствует.

Решение задачи

Диплом на то и диплом, что за 5 часов его так просто не напишешь. Поэтому вам даны целых 5 подсказок.

1. Добавить ребро и удалить ребро — сделать ребро «живым» на некотором отрезке времени.
2. Используйте метод «Разделяй и Властвуй».
3. Сжимайте компоненты двусвязности.
4. Даже когда компоненты двусвязности уже сжаты, если запросов мало, граф всё ещё можно сильно уменьшить.
5. Существует решение за $O(K \log K)$.

Формат входных данных

В первой строке записаны целые числа N и K : количество вершин и количество запросов, соответственно. $1 \leq N \leq 10^5$, $1 \leq K \leq 10^5$.

В следующих K строках по одному в строке записаны запросы. Каждый запрос начинается словом «ADD» или «DEL» для запроса на добавление или удаление ребра, соответственно. После этого слова записаны два целых числа a и b , задающих ребро. $1 \leq a, b \leq N$, $a \neq b$.

Формат выходных данных

После каждого запроса нужно вывести одно число — сколько в текущем графе мостов.

Примеры

bridges3.in	bridges3.out
4 8	1
ADD 1 2	2
ADD 2 3	0
ADD 1 3	2
DEL 2 3	1
DEL 1 2	2
ADD 2 4	3
ADD 1 4	0
ADD 2 3	

Задача В. Соединение и разъединение

Имя входного файла: `connect.in`
Имя выходного файла: `connect.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вы когда-нибудь слышали про обход в глубину? Например, используя этот алгоритм, вы можете проверить является ли граф связным за время $O(E)$. Вы можете даже посчитать количество компонент связности за то же время.

А вы когда-нибудь слышали про систему непересекающихся множеств? Используя эту структуру, вы можете быстро обрабатывать запросы “Добавить ребро в граф” и “Посчитать количество компонент связности в графе”.

А вы когда-нибудь слышали о *динамической* задаче связности? В этой задаче вам необходимо обрабатывать три типа запросов:

1. Добавить ребро в граф.
2. Удалить ребро из графа.
3. Посчитать количество компонент связности в графе.

Можно считать, что граф является неориентированным. Изначально граф является пустым.

Формат входных данных

В первой строке находятся два целых числа N и K — количество вершин и количество запросов, соответственно ($1 \leq N \leq 300\,000$, $0 \leq K \leq 300\,000$). Следующие K строк содержат запросы, по одному в строке. Каждый запрос имеет один из трех типов:

1. $+ u v$: Добавить ребро между вершинами u и v . Гарантируется, что такого ребра нет.
2. $- u v$: Удалить ребро между u и v . Гарантируется, что такое ребро есть.
3. $?$: Посчитать количество компонент связности в графе.

Вершины пронумерованы целыми числами от 1 до N . Во всех запросах $u \neq v$.

Формат выходных данных

Для каждого запроса типа ‘?’, Выведите количество компонент связности в момент запроса.

Примеры

<code>connect.in</code>	<code>connect.out</code>
5 11	5
?	1
+ 1 2	1
+ 2 3	2
+ 3 4	
+ 4 5	
+ 5 1	
?	
- 2 3	
?	
- 4 5	
?	

Задача С. Расстояние в дереве

Имя входного файла: `lenpath.in`
Имя выходного файла: `lenpath.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вам дано неориентированное дерево (связный граф без циклов) на N вершинах. Требуется отвечать на запросы вида «найти длину кратчайшего пути между данной парой вершин».

Формат входных данных

В первой строке находится одно целое число N ($1 \leq N \leq 10^5$) — количество вершин в дереве.

В каждой из следующих $N - 1$ строк задано по два числа, разделенных пробелом, u_i, v_i — вершины, соединенные i -ым ребром дерева ($1 \leq u_i, v_i \leq N$).

В следующей строке задано число M — количество запросов ($0 \leq M \leq 10^5$).

В каждой из следующих M строк задано по два числа — номера вершин, соответствующих очередному запросу.

Формат выходных данных

Выведите M строк, в каждой из которых содержится одно целое число — ответ на очередной запрос. Ответы необходимо вывести в порядке, соответствующем порядку запросов на входе программы.

Примеры

<code>lenpath.in</code>	<code>lenpath.out</code>
3	0
2 1	1
3 2	
2	
2 2	
2 1	

Задача D. Почтовая реформа

Имя входного файла:	mail.in
Имя выходного файла:	mail.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В Флатландии идет пора реформ. Недавно была проведена реформа дорог, так что теперь по дорогам страны из любого города можно добраться в любой другой, причем только одним способом. Также была проведена реформа волшебников, так что в каждом городе остался ровно один волшебник. Теперь же началась реформа почтовой системы.

Недавно образованное почтовое агентство «Экс-Федя» предлагает уникальную услугу — коллективную посылку. Эта услуга позволяет отправлять посылки жителям всех городов на каком-либо пути по цене обычной посылки. Удивительно, но пользоваться такой услугой стали только волшебники Флатландии, которые стали в большом количестве отправлять друг другу магические кактусы. Агентство столкнулось с непредвиденной проблемой: как известно, все волшебники живут в башнях и мало того, что не строят в них лестницы, так еще время от времени меняют их высоту. Поэтому, чтобы доставить посылку волшебнику, который живет в башне высотой h , курьеру агентства требуется иметь с собой не менее h метров веревки.

Вам поручено руководить отделом логистики — по имеющимся данным о высотах башен и об их изменениях вам нужно определять минимальную длину веревки, которую нужно выдать курьеру, который доставляет посылки между городами i и j .

Формат входных данных

Первая строка входного файла содержит число n — количество городов в Флатландии ($1 \leq n \leq 50\,000$). Во второй строке находится n положительных чисел, не превосходящих 10^5 — высоты башен в городах. В следующих $n - 1$ строках содержится по два числа u_i и v_i — описание i -й дороги, $1 \leq u_i, v_i \leq n, u_i \neq v_i$. В следующей строке содержится число k — количество запросов ($1 \leq k \leq 100\,000$). В следующих k строках содержатся описания запросов в следующем формате:

- Уведомление от волшебника из города i о том, что высота его башни стала равна h , имеет вид $! i h, 1 \leq i \leq n, 1 \leq h \leq 10^5$.
- Запрос от курьера о выдаче веревки для доставки посылок во все города на пути от i до j включительно имеет вид $? i j, 1 \leq i, j \leq n$.

Формат выходных данных

Для каждого запроса доставки посылок выведите минимальную длину веревки, которую необходимо выдать курьеру.

Примеры

mail.in	mail.out
3 1 2 3 1 3 2 3 5 ? 1 2 ! 1 5 ? 2 3 ! 3 2 ? 1 2	3 3 5
1 100 5 ! 1 1 ? 1 1 ! 1 1000 ? 1 1 ! 1 1	1 1000

Задача Е. Самая дальняя

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

Даны N точек на плоскости, нужно уметь обрабатывать следующие запросы:

- `get a b` — возвращает максимум по всем точкам величины $a \cdot x + b \cdot y$.
- `add x y` — добавить точку в множество.

Формат входных данных

Число N ($1 \leq N \leq 10^5$) и N точек. Далее число M ($1 \leq M \leq 10^5$) — количество запросов и собственно запросы. Формат запросов можно посмотреть в примере. Все координаты точек и числа a, b — целые числа, по модулю не превосходящие 10^9 .

Формат выходных данных

На каждый запрос вида `get` выведите одно целое число — максимум величины $a \cdot x + b \cdot y$.

Примеры

<code>mostfar.in</code>	<code>mostfar.out</code>
3	1
0 0	0
1 0	1
0 1	1
10	4
get 1 1	4
get -1 -1	1
get 1 -1	1
get -1 1	
add 2 2	
add -2 -2	
get 1 1	
get -1 -1	
get 1 -1	
get -1 1	

Задача F. Строки в дереве

Имя входного файла: `treestr.in`
Имя выходного файла: `treestr.out`
Ограничение по времени: 5 секунды
Ограничение по памяти: 256 мегабайт

Дано дерево. Дерево — это связный граф без циклов. На каждом ребре дерева написана строчная латинская буква. Между каждыми двумя вершинами существует ровно один простой путь, то есть путь по рёбрам дерева, проходящий через каждую вершину не более одного раза. Каждому пути соответствует строка, которая получается, если идти по этому пути и читать буквы на рёбрах в порядке следования. Путь можно проходить, начиная с любого его конца.

Также дана строка S . Соответствует ли она какому-либо простому пути в данном дереве? Длина строки и размер дерева не превышают $3 \cdot 10^5$.

Формат входных данных

Первая строка содержит заданную строку s . Следующая строка содержит количество вершин в дереве n . Следующие $n - 1$ строк описываются ребра деревьев в виде u, v, c , где u и v — вершины дерева, а c — символ, написанный на ребре.

Формат выходных данных

В первой строке выведите YES, если такой путь существует, и NO в противном случае.

Если путь существует, то выведите пару вершин, путь между которыми образует заданную строку.

Примеры

<code>treestr.in</code>	<code>treestr.out</code>
<code>abc 4 1 2 c 4 3 a 2 4 b</code>	<code>YES 1 3</code>
<code>abc 1</code>	<code>NO</code>
<code>zy 6 1 2 x 1 3 y 1 4 z 3 5 a 4 6 b</code>	<code>YES 4 3</code>

Замечание

В первом примере строка `abc` соответствует пути 3–4–2–1.

Во втором примере в дереве нет ни одного ребра.

В третьем примере строка `zy` соответствует пути 3–1–4.