

Вариант 4

1) #деревья, #динамика на дереве

Саше снится страшный сон. Ему приснилось бинарное дерево поиска, но когда он проснулся то не смог вспомнить значения ключей для всех вершин, а лишь помнил то, как выглядело дерево. Помогите Саше восстановить дерево по его описанию.

По заданному описанию дерева, для каждого узла определить значение ключа, таким образом, чтобы заданное дерево представляло из себя бинарное дерево поиска по ключу.

Все ключи – это целые положительные числа ($1 \leq k \leq n$). Значения ключей, для любой пары вершин, различны.

Входные данные

Первая строка содержит n – количество вершин в дереве. Каждая из следующих n строк имеет вид: $l_i r_i$, где l_i – номер левого дочернего узла для вершины с номером i , r_i - номер правого дочернего узла.

Выходные данные

Выведите n целых чисел, где k_i число означает значение ключа для вершины номер i .

2) #сортировки, #бинарный поиск, #деревья, #поиск в глубину

Саше снова приснился страшный сон. И было в том сне большое бинарное дерево поиска. Но не дерево пугало Сашу, а то, что не получалось у него искать в этом дереве элементы. Сколько ни пробовал загадывать ключ и искать его в дереве, каждый раз приходил куда-то не туда. много загадывал ключей, и все одно и то же. Отчание стало охватывать Сашу, как вдруг он понял: каждый раз, когда он искал ключ, этого ключа не было в дереве, и при этом случалась ровно одна ошибка. "Не беда!" — подумал он, — "А посчитаю-ка я матожидание значения элемента, который находится при поиске данного ключа." Только он собрался это сделать, как, вдруг, проснулся.

Итак, вам дано *бинарное дерево поиска*, то есть дерево, у которого в каждой вершине записано некоторое число, называемое *ключом вершины*. Количество сыновей у каждой вершины дерева равно либо 0, либо 2. У внутренней вершины есть *левый сын*, то есть сын, у которого ключ меньше ключа текущей вершины, и *правый сын*, у которого ключ больше ключа текущей вершины. Так же вам задан набор *ключей поиска*, все из которых различны и отличаются от ключей вершин, имеющих в дереве. Ключ, лежащий в листе, объявляется *результатом поиска*. Достоверно известно, что в ходе этого поиска мы ровно один раз ошибемся в сравнении, то есть пойдём не туда куда надо, а дальше ошибаться не будем. Все возможные ошибки равновероятны, то есть рассматриваются все такие поиски, в которых происходит ровно одна ошибка. Ваша задача — для каждого ключа поиска найти математическое ожидание (среднее значение) результата поиска, при условии что в этом поиске происходит ровно одна ошибка. То есть, надо для набора путей, в которых содержится ровно одна ошибка поиска заданного ключа, посчитать среднее значение ключей, находящихся в листьях этих путей.

Входные данные

В первой строке задано нечетное целое число n ($3 \leq n < 10^5$) — количество вершин в дереве. В следующих n строках заданы описания вершин. На $(i + 1)$ -ой строке записано два целых числа, разделенных пробелами. Первое число — это номер родителя i -ой вершины, второе — это ключ, лежащий в i -ой вершине. В следующей строке записано целое число k ($1 \leq k \leq 10^5$) — количество ключей, для которых надо посчитать среднее значение результатов поиска с одной ошибкой. В следующих k строках записаны сами ключи, по одному в каждой строке. Все ключи вершин и все ключи поиска — это целые положительные числа, не превосходящие 10^9 . Все $n + k$ ключей различны. Все вершины нумеруются от 1 до n . Для корня дерева вместо номера вершины родителя будет задано число «-1» (без кавычек). Гарантируется, что задано корректное бинарное дерево поиска. Для каждой вершины, кроме корня, согласно её ключу может быть установлено, является она левым или правым сыном.

Выходные данные

Выведите k действительных чисел, являющихся матожиданиями ответов для ключей, заданных во входе. Ответ должен отличаться от правильного с относительной или абсолютной погрешностью не более 10^{-9} .

3) # деревья, #перестановки

Саша страдает синдромом Аспергера. Поэтому развлечения у него не такие, как у большинства нормальных людей. Сейчас, например, он очень увлечен следующей игрой: он выписывает перестановку чисел от 1 до n , после чего последовательно добавляет числа перестановки в несбалансированное бинарное дерево поиска (свойства дерева поиска данной задачи совпадают со свойствами дерева из предыдущей). При добавлении элемента в несбалансированное бинарное дерево происходит спуск от корня дерева к листьям. На каждом этапе спуска происходит сравнение добавляемого ключа с ключом текущей вершины.

Для каждой перестановки Сашу интересует суммарное число сравнений, которое было сделано при добавлении ее элементов в несбалансированное бинарное дерево поиска. К сожалению, для некоторых перестановок процесс добавления затягивается очень надолго. Зная перестановку, помогите Саше быстро найти интересующую его величину.

Входные данные

В первой строке задано n ($1 \leq n \leq 200000$) – количество элементов в перестановке.

Во второй строке через пробел записано n попарно различных чисел – элементы перестановки.

Выходные данные

Выведите суммарное число сравнений.

4) #бинарный поиск, #теория чисел

Как Вы уже могли заметить, Саша очень странный человек. У него есть две любимые цифры a и b . Виталий называет целое положительное число *хорошим*, если в десятичной записи этого числа используются только цифры a и b . Саша называет хорошее число *замечательным*, если сумма его цифр является хорошим числом.

Например, пусть у Саши любимые цифры 1 и 3, тогда число 12 — не является хорошим, а числа 13 или 311 являются. Также число 111 — замечательное, а число 11 — нет.

Теперь Саша интересуется, сколько существует замечательных чисел длины ровно n . Так как количество таких чисел может быть довольно большим, он просит Вас посчитать остаток от деления этого количества на 1000000007 ($10^9 + 7$).

Под длиной числа подразумевается количество цифр в его десятичной записи без лидирующих нулей.

Входные данные

В первой строке записаны три целых числа: a, b, n ($1 \leq a < b \leq 9, 1 \leq n \leq 10^6$).

Выходные данные

Выведите единственное целое число — ответ на задачу по модулю 1000000007 ($10^9 + 7$).

5) #системы непересекающихся множеств, #деревья, #сортировки, #жадные алгоритмы

На доске нарисован граф-дерево из n вершин. Напомним, что неориентированный граф называется деревом, если он связан и не содержит циклов.

Каждая вершина графа покрашена в черный или красный цвет таким образом, что не существует двух вершин одного цвета, соединенных ребром. На каждом ребре записана его стоимость — неотрицательное целое число.

Мальчик Саша подошел к доске и написал около каждой вершины v число s_v — сумму стоимостей всех инцидентных этой вершине ребер, после чего он стер ребра и их стоимости с доски.

Ваша задача — восстановить исходное дерево по цветам вершин и числам s_v .

Входные данные

В первой строке входных данных записано единственное целое число n ($2 \leq n \leq 10^5$) — количество вершин в дереве. Далее в n строках записаны пары разделенных пробелом целых чисел c_i, s_i ($0 \leq c_i \leq 1, 0 \leq s_i \leq 10^9$), где c_i означает цвет i -ой вершины (0 — белый, 1 — черный), а s_i означает сумму стоимостей инцидентных i -ой вершине ребер в нарисованном на доске дереве.

Выходные данные

Выведите описание $n - 1$ ребер графа-дерева. Каждое описание — это тройка чисел v_i, u_i, w_i ($1 \leq v_i, u_i \leq n, v_i \neq u_i, 0 \leq w_i \leq 10^9$), где v_i и u_i — номера вершин, которые соединяет i -ое ребро, а w_i — его стоимость. Обратите внимание, что должно выполняться условие $c_{v_i} \neq c_{u_i}$.

Гарантируется, что для любых входных данных существует по крайней мере один соответствующий этим данным граф. Если существует несколько решений, то выведите любое. Ребра разрешается выводить в любом порядке. Числа при выводе разделяйте пробелами.

6) #поиск

Последовательность называется *отсортированной*, если она является неубывающей или невозрастающей. Например, последовательности [3, 1, 1, 0] и [1, 2, 3, 100] — отсортированы, но последовательность [1, 3, 3, 1] — нет. Вам дана последовательность чисел. Ваша задача — найти ее кратчайшую подпоследовательность, не являющуюся отсортированной.

Подпоследовательность — это последовательность, которая получается из данной путем удаления нуля или более ее элементов.

Входные данные

В первой строке записано целое число n ($1 \leq n \leq 10^5$). Далее через пробел записано n целых чисел, по модулю не превосходящих 10^6 — заданная последовательность чисел.

Выходные данные

Если кратчайшей неотсортированной подпоследовательности не существует, выведите 0. Иначе выведите ее длину k , а затем k целых чисел от 1 до n включительно — индексы элементов последовательности, образующих кратчайшую неотсортированную подпоследовательность. Если решений несколько, выведите любое.

7) #приоритетные очереди, #пирамиды

Саша и Леша идут в кино. Сейчас в местном кинотеатре действует акция под названием «Выбери Зал». Условия акции таковы:

- посетитель сам выбирает зал, в который он пойдет;
- пусть в выбранном зале на данный момент x ($x > 0$) свободных мест, тогда билет в такой зал стоит x злотых.

Очередь из n посетителей уже выстроилась перед единственной билетной кассой кинотеатра. Саша и Леша еще не встали в очередь, но им уже интересно: какое максимальное и минимальное количество злотых может заработать администрация кинотеатра, если все n посетителей будут покупать билеты по этой акции?

Посетители покупают билеты по очереди, сначала первый в очереди, потом второй, и так далее до n -го.

Входные данные

В первой строке записано два целых числа n и m ($1 \leq n, m \leq 1000$) — количество посетителей в очереди и количество залов в кинотеатре соответственно. В следующей строке записано m целых чисел a_1, a_2, \dots, a_m ($1 \leq a_i \leq 1000$) — a_i обозначает количество свободных мест в i -ом зале перед началом продажи билетов.

Числа в строках разделяются пробелами. Гарантируется, что суммарное количество свободных мест не меньше, чем n .

Выходные данные

Выведите два целых числа — максимальное и минимальное количество злотых, которое может заработать администрация кинотеатра соответственно.

8) #сортировки, #дерево отрезков, #дерево Фенвика

Девочка очень любит задачи про запросы на массиве.

Однажды ей попалась довольно известная задача: дан массив из n элементов (элементы массива проиндексированы от 1); также есть q запросов, каждый из которых задается парой целых чисел l_i, r_i ($1 \leq l_i \leq r_i \leq n$). Для каждого запроса необходимо найти сумму всех элементов массива с индексами от l_i до r_i включительно.

Такая задача показалась Девочке довольно скучной. Она решила, что перед тем, как отвечать на запросы, она перемешивает элементы массива, причем так, чтобы сумма ответов на все запросы была максимально возможной. Ваша задача — найти значение этой максимальной суммы.

Входные данные

Первая строка содержит два целых числа n ($1 \leq n \leq 2 \cdot 10^5$) и q ($1 \leq q \leq 2 \cdot 10^5$), разделенных пробелом — количество элементов в массиве и количество запросов, соответственно.

Следующая строка содержит n целых чисел a_i ($1 \leq a_i \leq 2 \cdot 10^5$), разделенных пробелами — элементы массива.

Каждая из следующих q строк содержит два целых числа l_i и r_i ($1 \leq l_i \leq r_i \leq n$), разделенных пробелом — i -тый запрос.

Выходные данные

В единственной строке выведите целое число — максимальную сумму ответов на запросы после перемешивания элементов массива.

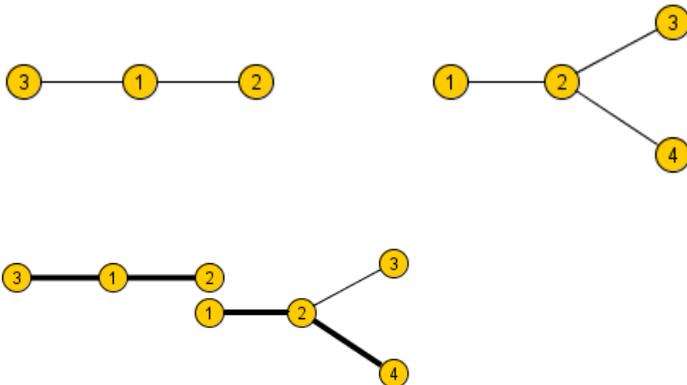
9) #деревья, #жадные алгоритмы, #поиск в ширину

Однажды мама попросила Сашу разобрать игрушки и избавиться от части из них. Саша нашел целую коробку игрушечных пауков. Они очень ему дороги, и мальчик не хочет их выкидывать. Саша придумал хитрый план: он склеит всех паучков и подвесит к потолку. Также он знает, что чем ниже будет свисать эта конструкция, тем больше понравится маме, и она не станет выкидывать его самые любимые игрушки. Помогите Саше осуществить его план.

Паук состоит из k бусинок, связанных $k - 1$ ниточкой. Каждая ниточка соединяет две различные бусинки, при этом любая пара бусинок, составляющих паука, либо непосредственно связана ниточкой, либо связана некоторой цепочкой из ниточек и бусинок.

Склеивать пауков Саша может непосредственно за бусинки. Длина каждой нитки равна 1. Размерами бусинок можно пренебречь. Поэтому можно считать, что склеивание пауков происходит путем отождествления некоторых бусинок (см. рисунок). При этом конструкция, получаемая при склеивании, также должна представлять собой паука, т.е. для нее должны выполняться указанные свойства.

После того, как Саша склеит всех пауков, он считает длину получившейся поделки. Расстояние между парой бусинок вычисляется как длина ниточек, соединяющих эти две бусинки. Длиной полученной конструкции называется наибольшее расстояние между всеми парами бусинок. Саша хочет получить конструкцию как можно большей длины.



Входные данные

В первой строке входного файла записано одно целое число n ($1 \leq n \leq 100$) — количество пауков. Следующие n строк содержат описания каждого паука: целое число n_i ($2 \leq n_i \leq 100$) — количество бусин, далее $n_i - 1$ пара чисел, означающих номера бусин, соединенных нитками. Бусинки, составляющие каждого паука, нумеруются от 1 до n_i .

Выходные данные

Выведите одно число — длину искомой конструкции.

10) #деревья, #динамика на дереве

А победил В. Эта победа была так важна, что А решил вырезать на скале сцену поражения В. Теперь ему нужно было найти наилучшее место и увековечить там свою победу!

Он решил посетить все n городов страны чтобы найти наиболее подходящую скалу. Но недавняя война утомила его, и он не хотел много путешествовать. Поэтому он хотел посетить каждый из этих n городов хотя бы один раз. При этом пройденное расстояние должно быть как можно меньше. Города соединены двусторонними дорогами. Из каждого города можно дойти до любого другого, причем единственным способом.

Все города пронумерованы от 1 до n . А сейчас в городе 1, и он хочет посетить все остальные города, пройдя как можно меньше. А может закончить свое путешествие в любом городе.

Найдите расстояние, которое А пройдет.

Входные данные

Первая строка содержит одно целое число n ($1 \leq n \leq 10^5$) — количество городов.

Следующие $n - 1$ строк содержат по 3 целых числа x_i , y_i и w_i ($1 \leq x_i, y_i \leq n$, $0 \leq w_i \leq 2 \times 10^4$). x_i и y_i — номера двух городов, соединенных дорогой, а w_i — длина дороги.

Выходные данные

Выведите одно число — минимальную длину пути А.