

Вариант 6

1) #деревья

В городе объявлена эпидемия. Больницы города не справляются с наплывом больных. Администрацией было решено вызвать эксперта по борьбе с эпидемией – X. Но этот мизантроп отказывается работать в команде с кем-либо в имеющихся госпиталях и требует себе новый. Город надо спасать, поэтому его требование решено было удовлетворить и новый госпиталь построить. Теперь необходимо выбрать место, на котором он будет построен.

Город представляет из себя n площадей, некоторые из которых соединены дорогами. Причем, от любой площади до любой другой можно доехать единственным способом. На i площади живет a_i человек. После открытия госпиталя, все люди пойдут в день открытия в этот госпиталь. Это учитывается при выборе места для строительства, для прохода с каждой стороны в госпитале будет своя дверь. Администрация не хочет, чтобы людям показалось, что будут огромные очереди, и поэтому, они хотят минимизировать длину самой большой очереди ко входу. Помогите выбрать такое место для строительства госпиталя, чтобы самая длинная очередь была как можно короче.

Входные данные

В первой строке n ($0 < n < 100001$). Во второй строке заданы n чисел a_i ($0 < a_i < 10^9$) – населенность площадей. Далее в $n-1$ строке, заданы номера соединенных дорогой площадей.

Выходные данные

Выведите номер площади, на которой можно построить госпиталь.

2) #графы

Для популяризации хоккея и повышения мастерства команд Урала был организован Всеуральский турнир. Для участия в турнире были приглашены N хоккейных команд из городов Урала.

После первых двух туров, в каждом из которых каждая команда провела по одному матчу, оказалось, что команд слишком много. Организаторам турнира было решено допустить к дальнейшему участию только K команд, никакие две из которых не встречались в рамках первых двух туров.

Требуется найти набор из K команд, удовлетворяющих условиям, либо вывести сообщение о том, что это сделать невозможно.

Входные данные

В первой строке N ($1 < N < 100001$, N - четно). Последующие N строк содержат описания матчей. Описание каждого матча состоит из двух натуральных чисел, не превышающих N – номеров команд, игравших в матче. Первые $N/2$ из них соответствуют матчам первого тура, оставшиеся – матчам второго тура.

Выходные данные

Результат содержит либо 0 либо K различных чисел – номера отобранных команд.

3) #графы, #поиск в ширину

Вам поручили очень ответственное задание. Правительство области вручает Вам план постройки дорог между городами. По плану все дороги односторонние, но между двумя городами может быть больше одной дороги, возможно, в разных направлениях. Вам необходимо вычислить минимальное k , такое, что данный план является слабо k -связным.

Правительство называет план слабо k -связным, если выполнено следующие условие: для любых двух различных городов можно проехать от одного до другого, нарушив правила движения не более k раз. Нарушение правил – это проезд по существующей дороге в обратном направлении.

Входные данные

В первой строке даны два числа $1 < n < 301$, $0 < m < 10^5$ – количество городов и дорог в плане. В следующих m строках даны по два числа – номера городов, в которых начинается и заканчивается соответствующая дорога.

Выходные данные

Выведите минимальное k , такое, что данный план является слабо k -связным.

4) #графы, #поиск в глубину

Дорожная сеть города устроена довольно просто. Все дороги являются отрезками единичной длины с концами в точках с целыми координатами. Этот факт – своего рода достопримечательность города.

Недавно пришедший к власти мэр считает, что он тратит слишком много времени из дома в мэрию и обратно. Он решил построить несколько дорог так, чтобы этот путь был как можно короче. Естественно, новые дороги должны также являться единичными отрезками с концами в точках с целыми координатами.

Вам поручено вычислить минимальное количество дорог, которое придется построить для осуществления плана мэра.

Входные данные

Первая строка содержит количество городов. Далее содержится описание дорог, в виде четырех целых чисел $x_i y_i x_j y_j$. Все координаты находятся в первой четверти координатной плоскости и не далее точки (100,100).

Последняя строка содержит координаты дома мэра. Мэрия расположена в точке с координатой (0,0)

Выходные данные

Выведите количество дорог которое необходимо построить.

5) #деревья

Структура компании Б имеет иерархический вид, то есть может быть представлена в виде дерева. Рассмотрим представление этой структуры в следующем виде:

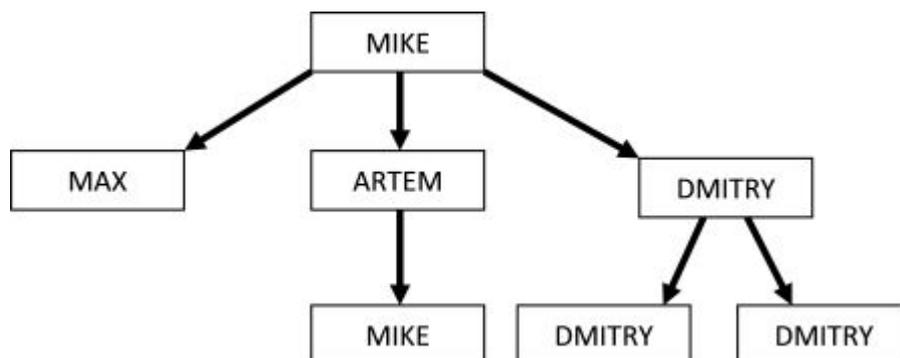
- $employee ::= name . | name : employee_1, employee_2, \dots, employee_k .$
- $name ::=$ имя сотрудника

То есть описание каждого сотрудника состоит из его имени, двоеточия, описаний всех его подчиненных, разделённых запятыми, и точки в конце. Если у сотрудника нет подчиненных, двоеточие в его описании отсутствует.

Например, строка `MIKE:MAX., ARTEM:MIKE., DMITRY:DMITRY., DMITRY...` является корректной записью структуры корпорации, в которой директор `MIKE` имеет подчиненных `MAX`, `ARTEM` и `DMITRY`. `ARTEM` имеет подчиненного, которого зовут `MIKE`, точно так же как и его начальника, а двоих подчиненных `DMITRY` зовут `DMITRY`, как и его самого.

В компании Б каждый сотрудник может переписываться только со своими подчиненными, причем не обязательно прямыми. Назовем неудобной ситуацию, когда человек с именем s пишет письмо другому человеку, которого так же зовут s . В приведенном выше примере есть 3 таких пары: одна с участием `MIKE`, и две для `DMITRY` (по одной на каждого его подчиненного).

Ваша задача — по заданной структуре корпорации найти количество неудобных пар в ней.



Входные данные

В первой и единственной строке записана структура компании — строка длиной от 1 до 1000 символов. Гарантируется, что описание корректно. Каждое имя — это строка из больших латинских букв длиной от 1 до 10 символов.

Выходные данные

Выведите одно число — количество неудобных ситуаций в компании.

6) #бинарный поиск, #сортировки, #жадные алгоритмы

Множество свободное от k -делимости — это множество целых чисел, в котором не существует двух целых чисел, таких, что первое число равно другому, умноженному на k . То есть, во множестве не существует двух целых чисел x и y ($x < y$), таких, что $y = x \cdot k$.

Вам дан набор, состоящий из n различных положительных целых чисел. Ваша задача — найти размер наибольшего его подмножества свободного от k -делимости.

Входные данные

Первая строка входных данных содержит два целых числа n и k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$). Следующая строка содержит список из n различных положительных целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Все числа в строках разделяются единичными пробелами.

Выходные данные

В единственной строке выходных данных выведите размер наибольшего подмножества свободного от k -делимости для $\{a_1, a_2, \dots, a_n\}$.

7) #бинарный поиск, #два указателя

Город D состоит из n башен, установленных последовательно на одной прямой. Высота i -той по порядку башни (слева направо) в последовательности равна h_i . Мэр города затеял перестройку, после которой город должен стать *красивым*. В *красивом* городе все башни расположены слева направо по неубыванию высот.

Перестройка состоит из выполнения нескольких (возможно нуля) операций. За одну операцию разрешается краном поднять любую башню и установить ее целиком на верх какой-то соседней башни. Другими словами, можно взять i -тую по порядку башню, и установить на верх либо $(i - 1)$ -ой (если такая существует), либо $(i + 1)$ -ой (если такая существует) по порядку башни. Высота получившейся башни равняется сумме высот соединенных башен. После этого две соединенные башни уже нельзя разъединить никаким образом, но с получившейся башней можно снова проводить подобные операции. Обратите внимание на то, что после выполнения одной операции общее количество башен, стоящих на прямой, уменьшается на 1.

Помогите мэру определить наименьшее количество операций, с помощью которых можно перестроить город в красивый.

Входные данные

Первая строка содержит единственное целое число n ($1 \leq n \leq 5000$) — количество башен в городе. Следующая строка содержит n целых чисел, разделенных пробелами: i -тое по порядку число h_i ($1 \leq h_i \leq 10^5$) обозначает высоту i -той по порядку башни (слева направо) в изначальной последовательности башен.

Выходные данные

Выведите единственное целое число — наименьшее количество операций, которое требуется, чтобы перестроить город в красивый.

8) #бинарный поиск

Евгений любит слушать музыку. В плейлисте Евгения играет n песен. Известно, что песня номер i имеет продолжительность t_i минут. Каждую песню плейлиста Евгений слушает, возможно, более одного раза. Песню номер i он слушает c_i раз. Плейлист Евгения построен следующим образом: сначала c_1 раз играет песня номер 1, потом c_2 раз играет песня номер 2, ..., в конце c_n раз играет песня номер n .

Евгений выписал себе на листочек m моментов времени, в которые ему понравились песни. Теперь для каждого такого момента, он хочет узнать номер песни, которая играла в этот момент времени. Момент времени x обозначает, что Евгений хочет знать, какая песня играла в x -ую от начала минуту прослушивания плейлиста.

Помогите Евгению, выведите требуемые номера песен.

Входные данные

В первой строке заданы два целых числа n, m ($1 \leq n, m \leq 10^5$). В следующих n строках заданы пары целых чисел. В i -ой строке записаны целые числа c_i, t_i ($1 \leq c_i, t_i \leq 10^9$) — описание плейлиста. Гарантируется, что суммарная продолжительность плейлиста не

более 10^9 ($\sum_{i=1}^n c_i \cdot t_i \leq 10^9$).

В следующей строке записаны m целых положительных чисел v_1, v_2, \dots, v_m , которые описывают времена, выписанные Евгением. Гарантируется, что не существует момента времени v_i , когда музыка уже не играла. Гарантируется, что $v_i < v_{i+1}$ ($i < m$).

Момент времени v_i обозначает, что Евгений хочет знать, какая песня играла в v_i -ую от начала минуту прослушивания плейлиста.

Выходные данные

Выведите m целых чисел — i -ое число должно быть равно номеру песни, которая играла в v_i минуту от начала прослушивания плейлиста.

9, 10) #дерево отрезков, #дерево Фенвика, #сортировки, #перестановки

Решите упражнения №5 и №6 параграфа 5.1.1 из книги «Искусство программирования» т.3 Д.Кнут.