

Задача А. Полотка Бамблби

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

После очередного боя с десептиконами у Бамблби что-то сломалось. Теперь он делает непонятные вещи. Например, сейчас он пытается решить некоторую странную задачу. Опишем ее условие.

Введем две функции, определенные на множествах положительных натуральных чисел. Первой функцией будет $\text{mex}(a_1, a_2, \dots, a_n)$ — наименьшее положительное натуральное число, которого нет среди чисел a_1, a_2, \dots, a_n . Второй функцией будет $\text{gcd}(a_1, a_2, \dots, a_n)$ — наибольший общий делитель чисел a_1, a_2, \dots, a_n . Бамблби сгенерировал массив натуральных чисел, а теперь хочет уметь делать следующую операцию: для произвольных чисел l и r перебирать все подмножества отрезка массива от a_l до a_r включительно, для каждого непустого подмножества считать его mex , а потом считать gcd всех получившихся mex -ов (несложно понять, что их может оказаться $2^{r-l+1} - 1$).

К сожалению, вычислительная мощность Бамблби не позволяет ему решать эту задачу быстро. Помогите ему.

Формат входного файла

В первой строке входного файла дано число n ($1 \leq n \leq 10^5$) — длина массива, сгенерированного Бамблби.

Во второй строке дано n разделенных пробелами чисел a_i ($1 \leq a_i \leq 10^9$) — числа в массиве, сгенерированном Бамблби.

В третьей строке входного файла дано число q — количество запросов ($1 \leq q \leq 10^4$).

В каждой из следующих q строк входного файла записаны два числа l, r ($1 \leq l \leq r \leq n$) — отрезок, на котором надо посчитать значение функции.

Формат выходного файла

В выходной файл выведите q строк — в i -ой строке выведите ответ на i -ый запрос.

Примеры

<code>bumblebee.in</code>	<code>bumblebee.out</code>
4	2
1 2 3 4	1
2	
1 1	
1 4	

Система оценивания

Первая группа тестов состоит из тестов, для которых выполняется ограничение $n \leq 4$. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 20 баллов.

Вторая группа тестов состоит из тестов, для которых выполняется ограничение $n \leq 1000$. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 40 баллов.

Третья группа тестов состоит из тестов, для которых выполняется ограничение $n \leq 10^5$. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 40 баллов.

Обратите внимание на возможность узнать результат проверки вашего решения на всех группах тестов, кроме последней, нажав на ссылку `Request feedback` на вкладке `Runs`.

Задача В. Штурм

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

Автоботы отважились на штурм базы десептиконов. Из-за эффекта внезапности множество десептиконов полегло на месте, а остальные начали судорожно прятаться в бункеры, которые начали закрываться. Оптимус Прайм хочет добить как можно больше десептиконов. Для этого он решил использовать новую разработку автоботов — бомбу «Антибункер».

Принцип работы бомбы заключается в том, что ее можно кинуть внутрь бункера, а когда тот закроется — взорвать, и тогда она убьет всех, кто был внутри. Единственное ее неудобство заключается в том, что взрывается она неавтоматически. То есть Оптимусу, после того, как он кинет бомбу, придется ждать закрытия бункера, чтобы взорвать всех внутри, и лишь потом он сможет поехать дальше.

Оптимус собирается проехать мимо каждого бункера ровно один раз, посетив их в порядке возрастания номеров. К счастью, он знает, на какой минуте закроется каждый из бункеров, а также количество десептиконов в каждом из бункеров. Между бункерами Оптимус перемещается очень быстро — перемещение между любой парой бункеров занимает у него ровно одну минуту. Если он проехал бункер, он уже не сможет вернуться обратно.

Оптимус Прайм просит вас помочь ему рассчитать, какое наибольшее количество врагов он сможет убить, и в какие бункеры нужно бросить бомбу для этого.

Формат входного файла

В первой строке дано число n ($1 \leq n \leq 10^5$) — количество бункеров.

Далее идут две строки по n целых чисел. В первой строке содержатся числа a_i ($1 \leq a_i \leq 10^9$) — количество десептиконов в бункере i . Во второй строке содержатся числа t_i ($1 \leq t_i \leq 10^5$) — время закрытия бункера i .

Формат выходного файла

В первой строке выходного файла выведите одно число — наибольшее возможное количество врагов, которые будут повержены Оптимусом. Во второй строке выведите количество бункеров, которое он сможет взорвать. Во третьей строке выведите через пробел номера бункеров, которые будут взорваны. Номера должны следовать по возрастанию. Из условия понятно, что момент закрытия каждого взорванного бункера должен быть больше, чем момент закрытия бункера, взорванного перед ним.

Примеры

<code>assault.in</code>	<code>assault.out</code>
3 1 2 1 1 2 3	4 3 1 2 3
5 4 1 5 9 3 9 7 9 8 8	10 2 2 4

Система оценивания

Первая группа тестов состоит из тестов, для которых выполняется ограничение $n \leq 1000$. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 60 баллов.

Первая группа тестов состоит из тестов, для которых выполняется ограничение $n \leq 10^5$. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 40 баллов.

Обратите внимание на возможность узнать результат проверки вашего решения на всех группах тестов, кроме последней, нажав на ссылку «Request feedback» на вкладке «Runs».

Задача С. Хэдмастеры

Имя входного файла: `headmasters.in`
Имя выходного файла: `headmasters.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Уже скоро должен состояться финальный бой между трансформерами. Все вокруг затихло и ждет последнего сражения.

Давайте рассмотрим то, как готовятся к бою хэдмастеры. N хэдмастеров решили расположиться на N целых точках числовой прямой с координатами $1, 2, \dots, N$. В каждой точке должен оказаться ровно один робот. Единственная загвоздка заключается в том, что M различных пар роботов должны быть соединены специальными кабелями. Кабели являются очень дорогостоящими, поэтому стратегически важно минимизировать их суммарную длину.

Если робот в точке с координатой x должен быть соединен с роботом, который находится в точке с координатой y , то для их соединения потребуется $|x - y|$ метров кабеля. Помогите хэдмастерам найти минимальное количество кабеля, которое необходимо потратить при оптимальном расположении роботов в указанных точках.

Формат входного файла

В первой строке входного файла записано два числа N ($2 \leq N \leq 20$) — количество хэдмастеров. Во второй строке находится одно целое число M — количество пар хэдмастеров, которые должны быть соединены. В следующих M строках заданы пары хэдмастеров, которые должны быть соединены. Пара задается ровно двумя натуральными числами, не превышающими N — номерами роботов. В каждой строке содержится ровно одна такая пара. Никакие две пары не совпадают.

Формат выходного файла

Выведите в первую строку выходного файла выведите единственное число — минимальное количество кабеля, которое придется потратить хэдмастерам.

Примеры

<code>headmasters.in</code>	<code>headmasters.out</code>
5	3
3	
1 2	
2 3	
4 5	

Комментарий

Одним из возможных оптимальных расположений роботов может быть следующий: 4, 5, 1, 2, 3.

Система оценивания

Первая группа тестов состоит из тестов, для которых выполняется ограничение $N \leq 10$. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 20 баллов.

Первая группа тестов состоит из тестов, для которых выполняется ограничение $N \leq 16$. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 40 баллов.

Третья группа тестов состоит из тестов, для которых выполняется ограничение $N \leq 20$. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 40 баллов.

Обратите внимание на возможность узнать результат проверки вашего решения на всех группах тестов, кроме последней, нажав на ссылку `Request feedback` на вкладке `Runs`.

Задача D. Прогулка

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

Устав от постоянных войн, Бамблби решил прокатиться по своему городу и посмотреть достопримечательности. В его навигационной системе город представлен в виде связанного графа, содержащего ровно n вершин и $n - 1$ ребро. Каждой вершине графа соответствует некоторая площадь в городе. Между некоторыми площадями существуют двусторонние дороги — ребра в графе. Известно, что от любой площади города можно добраться до любой другой, проехав при этом только по дорогам.

Про каждую дорогу известно, какое время Бамблби тратит на проезд по ней. Бамблби хочет потратить на прогулку ровно T единиц времени. Кроме этого, он хочет объехать ровно K различных площадей, побывав на каждой не более одного раза. Помогите ему выбрать 2 соответствующие площадям вершины так, чтобы путь между ними состоял ровно из $K - 1$ различных дорог, а время, затраченное на поездку, было бы равно T .

Бамблби тратит время только на перемещения по дорогам, суммарное время его присутствия на площадях равно нулю.

Формат входного файла

В первой строке входного файла задано три числа n , T и K ($1 \leq n \leq 100\,000$, $0 \leq K \leq 100\,000$, $0 \leq T \leq 1\,000\,000\,000$) — количество площадей, необходимое время поездки и необходимое количество дорог, участвующих в поездке.

Следующие $n - 1$ строк содержат по три числа a_i , b_i и t_i ($1 \leq a_i, b_i \leq n$, $1 \leq t_i \leq 10000$) — описание очередной дороги. Первые два числа являются номерами площадей, соединенных этой дорогой, а третье — временем поездки по ней.

Формат выходного файла

Выведите ответ на задачу. Если таких вершин не существует, выведите 00. Числа ответа необходимо упорядочить по возрастанию.

Примеры

road.in	road.out
5 3 2 1 2 1 2 3 1 3 5 1 2 4 2	1 4
2 0 0 1 2 10	1 1
2 1 0 1 2 10	0 0

Комментарий

В случае, если вариантов ответа несколько, выведите лексикографически минимальную пару.

Система оценивания

Первая группа тестов состоит из тестов, для которых выполняется ограничение $n \leq 100$. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 20 баллов.

Вторая группа тестов состоит из тестов, для которых выполняется ограничение $n \leq 1000$. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 20 баллов.

Третья группа тестов состоит из тестов, для которых выполняется ограничение $n \leq 100000$. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 60 баллов.

Обратите внимание на возможность узнать результат проверки вашего решения на всех группах тестов, кроме последней, нажав на ссылку «Request feedback» на вкладке «Runs».