

Задача А. Ох уж эти волонтеры

Имя входного файла: `volunteers.in`
Имя выходного файла: `volunteers.out`

Проведение олимпиады это всегда тяжело и основной рабочей силой являются волонтеры. Каждый волонтер находится в подчинении ровно у одного другого волонтера, кроме Амана, главного волонтера имеющего номер 1. Сделано это для того, чтобы по возможности максимально правильно распределять просьбы участников между ними.

Более формально: пусть к волонтеру под номером V пришло W просьб(принести воду, отнести вещи, и т.д.).

- Если у волонтера нет подчиненных, он выполняет все просьбы сам.
- Если у него есть K ($K > 0$) подчиненных, то в случае если W делится на K , волонтер V распределяет свою работу между своими подчиненными передавая каждому из них W/K просьб и те в свою очередь начинают действовать по тому же алгоритму.
- В противном случае, если W не делится на K , то все W просьб не выполняются.

Аману очень не нравится то что некоторые просьбы остаются не выполненными. Он хотел попросить прощения у участников, но для начала ему хотелось бы узнать общее количество просьб которые остались не выполненными.

Он обратился с этой задачей к своему хорошему другу НурлашКО. А он в свою очередь решил перекинуть эту задачу на участников.

Формат входных данных

Первая строка входного файла содержит количество волонтеров на олимпиаде N .

В следующих $N - 1$ строках идут пары чисел v_i, u_i означающие то что волонтер u_i является подчиненным волонтера v_i ($1 \leq v_i, u_i \leq N$).

В $N + 1$ строке содержится число M - количество запросов.

Далее в M строках идут описания запросов в виде пар V_i, W_i ($1 \leq V_i \leq N, 1 \leq W_i \leq 10^6$). W_i — общее количество просьб, V_i — номер волонтера.

Формат выходных данных

Выведите M строк. В i -ой строке выведите количество не выполненных просьб для i -го запроса.

Примеры

| <code>volunteers.in</code> | <code>volunteers.out</code> |
|----------------------------|-----------------------------|
| 5 | 5 |
| 1 2 | 0 |
| 1 3 | |
| 2 4 | |
| 2 5 | |
| 2 | |
| 1 10 | |
| 1 20 | |

Система оценки

Данная задача содержит 3 подзадачи:

1. $1 \leq N, M \leq 1\,000$. Оценивается в 17 баллов.
2. $1 \leq N, M \leq 100\,000$. Все запросы делаются к волонтеру номер 1. Оценивается в 24 баллов.
3. $1 \leq N, M \leq 100\,000$. Оценивается в 59 балла.

Каждая следующая подзадача оценивается только при прохождении всех предыдущих.

Задача В. Лжец

Сергазы и Улугбек любят играть в игры. Эта задача про их любимую игру.

Обозначим $N = \{0, 1, \dots, n-1\}$. Другими словами, N — это множество, состоящее из целых чисел от 0 до $n-1$ включительно. В начале игры Сергазы выбирает целое число x из множества N , но его выбор неизвестен Улугбеку. Цель Улугбека — найти такое подмножество N , что оно состоит из p различных элементов и **не содержит** x . Обозначим это подмножество за P . Для того, чтобы найти P , Улугбек может задавать Сергазы вопросы. В каждом из вопросов Улугбек выбирает множество S , которое является непустым подмножеством N , и спрашивает: «Принадлежит ли x множеству S ?». Сергазы отвечает на каждый вопрос либо «да», либо «нет».

Сергазы любит усложнять жизнь Улугбеку, так что на каждый вопрос он может либо солгать, либо сказать правду. Например, пусть $x = 1$. Если Улугбек спросит про подмножество $S = \{0, 2\}$, Сергазы может солгать, ответив "да" или он может сказать правду, ответив "нет". К счастью, Сергазы хороший друг, и он обещал Улугбеку, что он никогда не будет врать ему k раз подряд. То есть, среди k любых **последовательных** ответов будет как минимум 1 правдивый ответ.

Сергазы не просто лжец, но и к тому же хитрый лжец. Он может менять загаданное им число по ходу игры, но таким образом, что новое загаданное число не противоречит его ответам на предыдущие вопросы.

Кроме этого, Сергазы ответит максимум на q вопросов. Улугбек не хочет проигрывать, поэтому он просит у Вас помощи!

Пример

Предположим $n = 10$, $k = 2$, $p = 1$, и Сергазы загадал $x = 3$. Это означает, что Сергазы не будет давать $k = 2$ неправдивых ответов подряд, а Улугбек должен найти такое подмножество P , состоящее из 1-ого элемента, которое не содержит загаданное x .

Улугбек: $S = \{9, 8, 7\}$

Сергазы: (ложь)

Улугбек: $S = \{3, 9\}$

Сергазы: (правда)

Улугбек: $P = \{4\}$

Примечание: после двух вопросов Улугбек может быть уверен, что число, которое Сергазы загадал, не равняется 4.

Протокол взаимодействия

Даны n , k , p , q и функция, чтобы задавать вопросы, найдите P . Параметры n , p , k и q выбраны таким образом, что возможно найти такое множество P , которое удовлетворяет условию задачи. Вам нужно реализовать следующую функцию в вашем решении:

$player_answer(n, k, p, P)$ — эта функция будет вызвана грейдером ровно 1 раз

n : размер множества N .

k : количество подряд идущих вопросов, среди которых будет как минимум 1 правдивый ответ.

p : необходимый размер множества P .

P : массив целых чисел длины p ; как только вы найдете P , вы должны записать его элементы в P_0, \dots, P_{p-1} . Каждый элемент из P должен быть от 0 до $n-1$ включительно.

Эта функция не должна возвращать никакого значения.

Примечание: значение q не является параметром функции $player_answer$. Значение q вычисляется по формуле, которая отличается для каждой подзадачи. Для деталей, смотрите секцию подзадач.

Ваша программа может задавать Сергазы вопросы, используя следующую функцию:

$get_answer(S, sz)$ S : непустой массив различных чисел, которые составляют интересующее вас множество S . Каждый элемент множества S должен лежать между 0 и $n-1$ включительно. Примечание: вызов функции get_answer может изменять состав массива S . sz : размер множества S . Элементы множества, которые вы хотите спросить, должны быть записаны в S_0, \dots, S_{sz-1} . Функция

возвращает или 1, которое обозначает "да" или 0, что значит "нет в зависимости от того какой ответ сгенерировала программа жюри. Гарантируется, что среди любых k последовательных запросов будет дан как минимум 1 правдивый ответ.

Система оценки

Задача содержит 5 подзадачи.

1. $n = 4, k = 2, p = 1, q = 2$. Эта подзадача оценивается в 9 баллов.
2. $4 \leq n \leq 10^4, k = 2, 1 \leq p \leq n - 3, q = 2 \cdot p$. Эта подзадача оценивается в 10 баллов.
3. $2^k \leq n \leq 10^6, 2 \leq k \leq 18, p = 1, q = k$. Эта подзадача оценивается в 17 баллов.
4. $2^k \leq n \leq 10^6, 2 \leq k \leq 18, 1 \leq p \leq n - 2^k + 1, q = k \cdot p$. Эта подзадача оценивается в 18 баллов.
5. $3 \cdot 2^k \leq n \leq 10^6, 2 \leq k \leq 18, 1 \leq p \leq n - 3 \cdot 2^k + 1, q = \max(\lfloor \frac{k \cdot p}{2} \rfloor, k)$. Эта подзадача оценивается в 46 баллов.

Примечание: $\lfloor x \rfloor$ обозначает наибольшее целое число, которое не больше, чем x . Так же есть дополнительные ограничения для всех подзадач: $n \cdot p \leq 4 \cdot 10^6$.

Если ваша программа попытается задать больше вопросов, чем разрешено, она будет немедленно завершена с вердиктом "неправильный ответ".

Пример грейдера

Грейдер считывает данные в следующем формате:

Строка 1: n, k, p, q

Строка 2: x

Грейдер выводит ваш ответ и количество запросов, которые вы задали, а также проверяет, корректен ли ваш ответ.

Задача С. Хорошие отрезки

Имя входного файла: `segments.in`
Имя выходного файла: `segments.out`

Дан массив a длины N , состоящий из целых положительных чисел от 1 до N , включительно. Требуется посчитать количество **хороших** отрезков в этом массиве (два отрезка являются различными если у них различается хотя бы один из концов). Отрезок $[l, r]$ называется хорошим, если для любого числа x , которое в нем встречается, количество его вхождений в отрезок — ровно x . Например, отрезок из чисел $(2, 1, 4, 4, 2, 4, 4)$ — хороший, а отрезок $(3, 3, 2, 2, 3, 3)$ — нет (число 3 встречается 4 раза).

Формат входных данных

Первая строка входного файла содержит целое число N ($1 \leq N \leq 500\,000$). Во второй строке содержатся N целых чисел a_1, a_2, \dots, a_N ($1 \leq a_i \leq N$).

Формат выходных данных

В единственную строку выходного файла выведите одно число — количество хороших отрезков.

Примеры

| <code>segments.in</code> | <code>segments.out</code> |
|--------------------------|---------------------------|
| 8 2 2 1 4 4 2 4 4 | 4 |

Система оценки

Данная задача содержит восемь подзадач:

- $1 \leq N \leq 200$. Оценивается в 12 баллов.
- $1 \leq N \leq 500$. Оценивается в 13 баллов.
- $1 \leq N \leq 2000$. Оценивается в 12 баллов.
- $1 \leq N \leq 5000$. Оценивается в 13 баллов.
- $1 \leq N \leq 20\,000$. Оценивается в 12 баллов.
- $1 \leq N \leq 50\,000$. Оценивается в 13 баллов.
- $1 \leq N \leq 200\,000$. Оценивается в 12 баллов.
- $1 \leq N \leq 500\,000$. Оценивается в 13 баллов.

Каждая подзадача оценивается только при прохождении всех предыдущих.⁴