

Задача 8. Выполнить план, но не перевыполнить

Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

Это интерактивная задача.

Требуется разработать планы производства автомобилей и запасных частей в некоторой стране. Всего в стране n городов, в которых расположены заводы, задействованные в производстве. В i -м городе на заводе могут работать от l_i до r_i человек включительно.

Некоторые города соединены двусторонними дорогами, при этом дорожная сеть имеет форму дерева: от каждого города можно единственным способом добраться до любого другого города, не проезжая через один город дважды.

Планом производства будем называть последовательность целых чисел $[a_1, a_2, \dots, a_n]$, где a_i — количество человек, которые будут работать на i -м заводе ($l_i \leq a_i \leq r_i$). После формирования плана производства некоторые заводы будут выбраны как *сборочные*, они будут производить автомобили, а остальные будут производить запчасти. Выбор считается *рациональным*, если никакие два сборочных завода не соединены дорогой напрямую. Среди всех возможных рациональных выборов будет выбран тот, для которого суммарное количество работников сборочных заводов будет максимально. Это число называется *эффективностью* плана $[a_1, a_2, \dots, a_n]$.

В этой задаче для некоторых значений v_1, v_2, \dots, v_q вам необходимо выяснить, существует ли план с эффективностью v_i . Если такой план существует, вам может потребоваться предъявить такой план.

Зафиксированы целочисленные параметры x, y и m . Рассмотрим план $[a_1, a_2, \dots, a_n]$. Сертификатом этого плана назовем число $k = \bigoplus_{j=1}^n ((x \cdot j + y \cdot a_j^2) \bmod m)$, где \bigoplus — это операция «побитового исключающего или».

Напомним, что эта операция обозначается «xor» в Паскале и Python, «^» в C++ и Java; для двух целых чисел она определена следующим образом: i -й бит результата равен 1 тогда и только тогда, когда в одном из чисел этот бит 1, а в другом 0. Например, $6 \oplus 10 = 110_2 \oplus 1010_2 = 1100_2 = 12$.

Процесс составления планов будет состоять из двух этапов.

На первом этапе вам будут даны значения v_1, v_2, \dots, v_q . Для каждого из них вам необходимо выяснить, существует ли план с эффективностью v_i и, если его не существует, вывести для этого запроса -1 , а если существует, то неотрицательное целое число k_i .

На втором этапе некоторые планы будут проверены: c раз вам будет дано целое число i ($1 \leq i \leq q$). В ответ на такой запрос требуется либо вывести -1 , если плана с эффективностью v_i не существует, либо предоставить план $[a_1, a_2, \dots, a_n]$, сертификат которого равен k_i , а эффективность равна v_i .

Предоставление сертификатов составленных планов и последующая проверка будут реализованы в интерактивном режиме. До того, как вы предоставите сертификаты, вы не будете знать, какие планы будут проверены. Поэтому в тех подзадачах, в которых $c > 0$, необходимо еще на первом этапе подготовиться к проверке, выведя для значений эффективности v_i , где искомым план существует, такие значения k_i , для которых вы сможете на втором этапе предъявить план с сертификатом, равным k_i .

Протокол взаимодействия

Это интерактивная задача. Ваша программа будет взаимодействовать с программой жюри с использованием стандартных потоков ввода и вывода. Каждое взаимодействие будет состоять из решения задачи для нескольких наборов входных данных.

Сначала ваша программа должна считать целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных в рамках одного взаимодействия с программой жюри. Затем t раз необходимо выполнить взаимодействие по решению задачи для набора входных данных.

Рассмотрим протокол взаимодействия для одного набора входных данных.

Сначала ваша программа должна считать данные в следующем формате.

В первой строке находится два целых числа n, q ($2 \leq n \leq 2 \cdot 10^5, 1 \leq q \leq 2 \cdot 10^5$) — количество городов и количество планов, которые надо составить.

Во второй строке находится три целых числа x, y, m ($11 \leq m \leq 2^{30}$; $0 \leq x, y < m$) — параметры вычисления сертификатов планов.

В следующих $n - 1$ строках находится описание дерева дорожной сети между городами. В i -й из этих строк находятся два целых числа s_i, f_i ($1 \leq s_i, f_i \leq n$; $s_i \neq f_i$) — описание двусторонней дороги между городами s_i и f_i . Гарантируется, что данные дороги образуют дерево.

i -я из следующих n строк содержит два целых числа l_i, r_i ($0 \leq l_i \leq r_i \leq 10^9$) — ограничения на количество работников в i -м городе.

В следующей строке находятся q целых чисел v_1, v_2, \dots, v_q ($0 \leq v_i \leq \sum_{i=1}^n r_i$) — значения эффективности планов, которые нужно составить. Гарантируется, что все числа v_i различны.

После того как вы считали описание набора входных данных, вы должны вывести q целых чисел k_1, k_2, \dots, k_q ($-1 \leq k_i < 2^{30}$), где $k_i = -1$, если план с эффективностью v_i невозможно составить, иначе $0 \leq k_i < 2^{30}$.

После этого может быть выполнена проверка некоторых из составленных планов.

Проверка выполняется в интерактивном режиме следующим образом. Программа жюри выводит одну строку, содержащую целое число i ($1 \leq i \leq q$) — номер плана для проверки. Гарантируется, что все номера запрошенных планов будут различными.

В ответ вы должны вывести -1 , если i -й план составить невозможно. В этом случае ранее должно быть выведено $k_i = -1$. Иначе необходимо вывести n целых чисел a_1, a_2, \dots, a_n ($l_i \leq a_i \leq r_i$) — составленный план. Сертификат этого плана должен быть равен k_i , эффективность должна быть равна v_i .

После $c \geq 0$ проверок программа жюри передаст на вход вашей программе число $i = 0$, что означает, что работа с этим набором входных данных закончена, и ваше решение должно начать решать следующий набор входных данных или завершить работу, если этот набор был последним.

Обозначим сумму значений n в одном тесте как N . Обозначим сумму значений q в одном тесте как Q . Гарантируется, что $N \leq 2 \cdot 10^5$; $Q \leq 2 \cdot 10^5$, сумма c по всем наборам входных данных не превосходит 10^4 , сумма $n \cdot c$ по всем наборам входных данных не превосходит 10^6 .

Поскольку задача интерактивная, после каждого вывода строки вашей программой выводите символ перевода строки и делайте сброс буфера потока вывода.

Если вы используете «`cout << ... << endl`» в C++, «`System.out.println`» в Java, «`print`» в Python, «`writeln`» в Паскале, то сброс потока вывода у вас происходит автоматически, дополнительно ничего делать не требуется. Если вы используете другой способ вывода, рекомендуется делать сброс буфера потока вывода. Обратите внимание, что перевод строки надо выводить в любом случае. Для сброса буфера потока вывода можно использовать «`fflush(stdout)`» в C++, «`flush(output)`» в Паскале, «`System.out.flush()`» в Java, «`sys.stdout.flush()`» в Python.

Система оценки

Подз.	Баллы	Ограничения			Необх. подзадачи
		n, N	Q	дополнительно	
1	11			$l_i = r_i$	
2	9			$c = 0$	
3	12			$l_i = 0, r_i \leq 1$	
4	4			$l_i = 0$	3
5	8			$s_i = 1,$ $f_i = i + 1$	
6	5	$n \leq 10,$ $N \leq 1000$	$Q \leq 1000$	$c = q, r_i \leq 10,$ $r_i - l_i \leq 2$	
7	2	$n \leq 10,$ $N \leq 1000$	$Q \leq 1000$	$c = q, r_i \leq 10$	6
8	13	$N \leq 1000$	$Q \leq 1000$	$c = q,$ $\sum_{i=1}^n r_i - l_i \leq 10^4$	6-7
9	11	$N \leq 1000$	$Q \leq 1000$	$c = q$	6-8
10	6			$c = q$	6-9
11	5	$N \leq 1000$			У, 6-9
12	5	$N \leq 8000$			У, 6-9, 11
13	9				У, 1-12

Замечание

В примерах сообщения программы участника и программы жюри разделены пустыми строками. Это сделано для облегчения восприятия, чтобы было понятно, какое сообщение является ответом на какое. В реальном взаимодействии в тестирующей системе пустых строк не будет.

Примеры

стандартный ввод	стандартный вывод
1	
9 3	
4 7 15	
1 2	
2 4	
2 5	
1 3	
3 6	
3 7	
6 8	
6 9	
4 4	
2 2	
5 5	
3 3	
2 2	
6 6	
3 3	
4 4	
3 3	
18 19 20	
1	-1 10 -1
2	-1
3	4 2 5 3 2 6 3 4 3
0	-1

стандартный ввод	стандартный вывод
3	
3 4	
3 4 11	
1 2	
2 3	
0 1	
0 1	
0 1	
0 1 2 3	
2	12 8 3 -1
0	1 0 0
4 6	
1 2 11	
1 2	
2 3	
3 4	
0 2	
1 1	
1 1	
1 2	
0 1 2 3 4 5	
5	-1 -1 4 14 9 -1
2	2 1 1 2
3	-1
0	1 1 1 1
5 7	
11 31 101	
1 2	
2 3	
2 4	
3 5	
1 2	
1 5	
0 4	
1 4	
4 6	
13 12 11 10 9 8 6	
3	-1 127 23 58 13 90 91
5	2 5 4 1 6
7	2 4 4 3 4
0	1 1 0 1 4

Пояснение к примеру

Первый тест подходит под ограничения подзадачи 1. В единственном наборе входных данных есть единственный способ составить план, это $a = [4, 2, 5, 3, 2, 6, 3, 4, 3]$. Его эффективность равна 19, а сертификат этого плана равен 10.

Во втором тесте три набора входных данных.

В первом наборе входных данных:

- Существует единственный план с эффективностью 0, для которого $a = [0, 0, 0]$. Сертификат этого плана равен 12.
- Существуют планы с эффективностью 1, например, $a = [1, 0, 0]$. Сертификат этого плана равен 8.
- Существуют планы с эффективностью 2, например, $a = [1, 0, 1]$. Сертификат этого плана равен 3.
- Не существует плана с эффективностью 3.

Среди четырех запрошенных планов был проверен план с номером $i = 2$ ($v_2 = 1$).

Во втором наборе входных данных:

- Не существует плана с эффективностью 0.
- Не существует плана с эффективностью 1.
- Существуют планы с эффективностью 2, например $a = [1, 1, 1, 1]$. Сертификат этого плана равен 4.
- Существуют планы с эффективностью 3, например $a = [2, 1, 1, 1]$. Сертификат этого плана равен 14.
- Существует единственный план $a = [2, 1, 1, 2]$ с эффективностью 4. Сертификат этого плана равен 9.
- Не существует плана с эффективностью 5.

Среди шести запрошенных планов были проверены планы с номерами $i = 5$ ($v_5 = 4$), $i = 2$ ($v_2 = 1$), $i = 3$ ($v_3 = 2$).

В третьем наборе входных данных были составлены следующие планы (начиная со второго, так как плана с эффективностью $v_1 = 13$ не существует): $[2, 5, 4, 4, 6]$; $[2, 5, 4, 1, 6]$; $[2, 4, 4, 4, 4]$; $[2, 4, 4, 3, 4]$; $[2, 4, 4, 2, 4]$; $[1, 1, 0, 1, 4]$.