

Задача А. Настойки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В линию выстроены n настоек, причем настойка 1 находится слева, а настойка n — справа. Каждая настойка увеличит ваше здоровье на a_i , если ее выпить. a_i может быть отрицательным, что означает, что настойка уменьшит ваше здоровье.

Вы начинаете с 0 здоровья и будете идти слева направо, от первой настойки до последней. Для каждой настойки вы можете выбрать, выпить ли ее. Вы должны следить за тем, чтобы ваше здоровье всегда было неотрицательным.

Какое наибольшее количество настоек вы можете выпить?

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 200\,000$) — количество настоек.

Следующая строка содержит n целых чисел $a_1, a_2, \dots, a_n$ ($-10^9 \leq a_i \leq 10^9$), которые обозначают изменения здоровья после употребления данных настоек.

Формат выходных данных

Выведите одно целое число — максимальное количество настоек, которое вы можете выпить, чтобы ваше здоровье всегда было неотрицательным.

Примеры

стандартный ввод	стандартный вывод
6 4 -4 1 -3 1 -3	5
5 1 3 -4 2 1	5
12 -3 -3 -7 -7 -1 -7 3 3 -2 -1 0 -7	5

Замечание

В примере, вы можете выпить 5 настоек, приняв настойки 1, 3, 4, 5 и 6. Невозможно выпить все 6 настоек, потому что в какой-то момент ваше здоровье станет отрицательным.

Задача В. Приключение

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Теплым весенним днем группа из N школьников-программистов гуляла в окрестностях города Кисловодска. К несчастью, они набрали на большую и довольно глубокую яму. Как это случилось — непонятно, но вся компания оказалась в этой яме.

Глубина ямы равна H . Каждый школьник знает свой рост по плечи h_i и длину своих рук l_i . Таким образом, если он, стоя на дне ямы, поднимет руки, то его ладони окажутся на высоте $h_i + l_i$ от уровня дна ямы. Школьники могут, вставая друг другу на плечи, образовывать вертикальную колонну. При этом любой школьник может встать на плечи любого другого школьника. Если под школьником i стоят школьники $j_1, j_2, \dots, j_k$, то он может дотянуться до уровня $h_{j_1} + h_{j_2} + \dots + h_{j_k} + h_i + l_i$.

Если школьник может дотянуться до края ямы (то есть $h_{j_1} + h_{j_2} + \dots + h_{j_k} + h_i + l_i \geq H$), то он может выбраться из нее. Выбравшиеся из ямы школьники не могут помочь оставшимся.

Найдите наибольшее количество школьников, которые смогут выбраться из ямы до прибытия помощи, и перечислите их номера.

Формат входных данных

В первой строке входного файла записано натуральное число N ($1 \leq N \leq 10^5$) — количество школьников, попавших в яму.

Далее в N строках указаны по два целых числа: рост i -го школьника по плечи h_i и длина его рук l_i ($1 \leq h_i, l_i \leq 10^9$).

В последней строке указано целое число — глубина ямы H ($1 \leq H \leq 10^9$).

Формат выходных данных

В первой строке выведите K — максимальное количество школьников, которые смогут выбраться из ямы. Если $K > 0$, то во второй строке выведите их номера в том порядке, в котором они вылезают из ямы. Школьники нумеруются с единицы в том порядке, в котором они заданы во входном файле. Если существует несколько решений, выведите любое.

Примеры

стандартный ввод	стандартный вывод
2 10 4 5 2 20	0
6 6 7 3 1 8 5 8 5 4 2 10 5 30	4 2 5 3 1

Задача С. Фейерверк

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Фейерверк — это одна из самых захватывающих частей фестиваля. Для фейерверка очень важно, чтобы все взрывные устройства, соединенные набором фитилей с точкой поджога, взорвались одновременно в нужный момент. Так как взрывные устройства, используемые для фейерверка, очень опасны, они установлены на некотором расстоянии от точки поджога, и соединены с ней фитилями. Чтобы соединить несколько взрывных устройств с точкой поджога, фитили соединяются наподобие ребёр дерева, как показано на Рисунке 1. Искра появляется на точке поджога и распространяется по фитилям. Когда искра достигает соединения нескольких фитилей, она начинает распространяться по всем ним. Скорость искры постоянна. Рисунок 1 показывает, как 6 взрывных устройств $\{E_1, E_2, \dots, E_6\}$ могут быть соединены, а также какой длины должен быть каждый использованный для этого фитиль. Кроме того, показано время взрыва каждого устройства, если считать, что искра в точке поджога появилась в момент времени 0

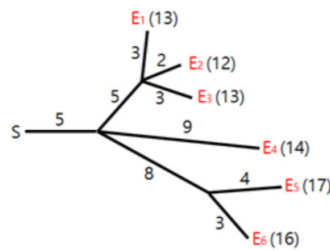


Рис. 1: Схема соединения

Хьюнмин должен подготовить схему соединения. К сожалению, в его схеме взрывные устройства могут взорваться не в одно и тоже время. Необходимо изменить длины некоторых фитилей, так чтобы все устройства взорвались в одно и тоже время. Например, чтобы все взрывные устройства на рисунке взорвались в момент времени 13 длины фитилей надо изменить как показано слева на Рисунке 2. Чтобы все взрывные устройства на Рисунке 1 взорвались в момент времени 14 длины фитилей надо изменить как показано слева на Рисунке 2.

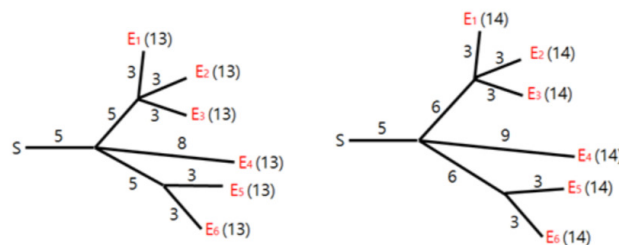


Рис. 2: Примеры изменения длин, после которых взрывы происходят одновременно

Стоимость изменения длины фитиля равна абсолютному значению разности длин до и после изменения. Например, стоимость изменения схемы на Рисунке 1 до схемы слева на Рисунке 2 равна 6. Стоимость изменения схемы на Рисунке 1 до схемы слева на Рисунке 2 равна 5.

Длину провода можно уменьшить до 0.

По заданной схеме соединения, необходимо изменить длины некоторых фитилей, так, чтобы все взрывные устройства взорвались одновременно, с минимальной стоимостью.

Формат входных данных

Все числа во входных данных целые положительные. Пусть N обозначает количество точек соединения фитилей, M — количество взрывных устройств ($1 \leq N+M \leq 300\,000$). Точки соединения фитилей пронумерованы от 1 до N . Точка поджога находится в соединении номер 1. Взрывные устройства пронумерованы от $N+1$ до $N+M$.

Входные данные даны в следующем формате:

N M
 P_2 C_2
 P_3 C_3
...
 P_N C_N
 P_{N+1} C_{N+1}
...
 P_{N+M} C_{N+M}

$P_i, 1 \leq P_i < i$, обозначает точку соединения фитилей, к которой проведен фитиль от точки соединения или взрывного устройства номер i . C_i обозначает длину фитиля, который их соединяет ($1 \leq C_i \leq 10^9$). В каждой точке соединения, кроме той, в которой расположена точка поджога, сходятся хотя бы 2 фитиля. С каждым взрывным устройством соединен ровно 1 фитиль.

Формат выходных данных

Выведите минимальную стоимость изменения длин фитилей, после которого все взрывные устройства взорвутся одновременно.

Примеры

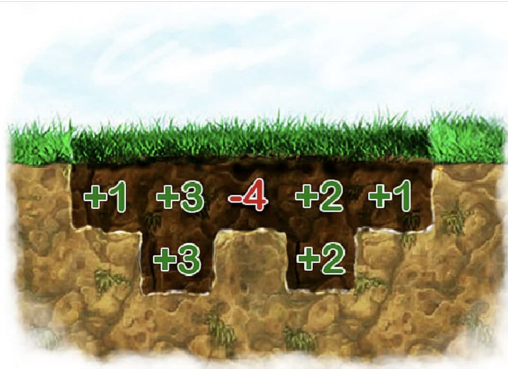
стандартный ввод	стандартный вывод
4 6 1 5 2 5 2 8 3 3 3 2 3 3 2 9 4 4 4 3	5

Задача D. Археологи

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Ваша команда охотников за сокровищами нашла огромные залежи сокровищ! Это можно представить как n мест для раскопок на прямой. Каждое из этих мест имеет свою доходность p_i , то есть за выкапывание каждого метра в глубину в этом месте ваша команда получает p_i юаней.

Конечно, вы хотите получить как можно больше юаней. Но чтобы не создавать оползней, после ваших раскопок не должно получиться слишком отвесных склонов. Формально, для любых двух соседних мест раскопок их глубина должна отличаться не больше чем на 1. В частности, глубина в 1-м и n -м местах должна быть 0 или 1. Какую максимальную прибыль можно получить?



Формат входных данных

Первая строка содержит целое положительное n ($1 \leq n \leq 250\,000$). Вторая строка содержит n целых чисел p_i ($-10^6 \leq p_i \leq 10^6$)

Формат выходных данных

Выведите одно число — максимальную прибыль, которую можно получить.

Примеры

стандартный ввод	стандартный вывод
5 1 3 -4 2 1	8
4 1 1 -2 3	5
5 -1 -3 0 -5 -4	0

Задача Е. Балансировка дерева

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Дано корневое дерево с N вершинами, корень дерева находится в вершине под номером 1. Каждое ребро, скажем i -тое, имеет два параметра, сопоставленные ему: C_i — стоимость ребра и параметр D_i .

Ваша задача — сделать равными стоимости всех путей из корня дерева в любой лист. Стоимость пути равна сумме всех стоимостей ребер на пути. Для того, чтобы решить задачу, Вы можете уменьшать или увеличивать стоимость какого-либо ребра. Для того чтобы сделать стоимость i -го ребра равной X , требуется $D_i \cdot |X - C_i|$ времени (в любой момент времени можно менять стоимость только одного ребра, т.е. все операции производятся последовательно, а не одновременно).

Найдите наименьшее время, требуемое для того, чтобы сделать равными стоимости всех путей из корня дерева в любой лист. Более того, выведите также новые стоимости ребер. Если существует более одного решения, то выведите любое из них.

Отметим, что вы можете сделать стоимости ребер отрицательными, но все они должны оставаться целыми.

Формат входных данных

Первая строка содержит единственное целое число T ($1 \leq T \leq 200\,000$) — количество тестовых случаев. Далее следует описание тестов в следующем формате:

Первая строка каждого теста содержит единственное целое число N ($1 \leq N \leq 200\,000$) — количество вершин в дереве. Каждая из следующих $N - 1$ строк содержит разделенные пробелами целые числа u_i, v_i, C_i, D_i ($1 \leq u_i, v_i \leq N$, $1 \leq C_i, D_i \leq 10^6$), задающие ребро с параметрами C_i и D_i , соединяющее вершины с номерами u_i и v_i .

Гарантируется, что сумма N по всем тестовым случаям не превосходит 200 000.

Формат выходных данных

Для каждого тестового случая выведите N строк. Первая строка должна содержать целое число — затраченное время. В каждой из следующих $N - 1$ строк i -я строка должна содержать новую стоимость i -го ребра.

Примеры

стандартный ввод	стандартный вывод
1	19
5	5
1 2 5 4	15
1 3 15 15	10
2 4 3 2	10
2 5 5 1	

Задача F. CookieClicker

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Костя играет в компьютерную игру Cookie Clicker, основная цель которой — накопление запасов печенья. Получать печенье можно с помощью различных *строений*: можно просто кликать по специальному полю на экране, получая печенье за клики, можно купить фабрику печеньев, алхимическую лабораторию, машину времени, и все это будет приносить много-много печеньев.

В начале игры (момент времени 0) у Кости есть 0 печеньев и ни одного строения. На его выбор доступно n строений: i -ое строение стоит c_i печеньев и в конце каждой секунды приносит v_i печеньев. Кроме того, чтобы играть было еще интереснее, Костя решил поставить себе ограничение: в каждый момент времени он будет использовать только одно строение. Разумеется, он может по своему усмотрению менять активное строение каждую секунду.

Важно, что в версии игры, в которую играет Костя, покупать новые строения и менять активное строение можно только в моменты времени, кратные одной секунде. В один момент времени Костя может купить новое строение и тут же начать его использовать. Если Костя начал использовать строение в момент времени t , то первую прибыль оно сможет принести лишь в момент времени $t+1$.

Костя хочет заработать не менее s печеньев как можно быстрее. Определите, за сколько секунд он сможет это сделать.

Формат входных данных

В первой строке находятся два целых числа n и s ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq s \leq 10^{16}$) — количество строений в игре и количество печеньев, которое Костя хочет заработать.

В каждой из следующих n строк записано два целых числа v_i и c_i ($1 \leq v_i \leq 10^8$, $0 \leq c_i \leq 10^8$) — количество печеньев, которое приносит i -ое строение за одну секунду, и стоимость этого строения.

Формат выходных данных

Выведите единственное целое число — минимальное количество секунд, за которое Костя сможет заработать не менее s печеньев. Гарантируется, что он может это сделать.

Примеры

стандартный ввод	стандартный вывод
3 9 1 0 2 3 5 4	6
3 6 1 0 2 2 5 4	5
3 13 1 0 2 2 6 5	7
1 10000000000000000 1 0	10000000000000000

Задача G. Не выделяйся

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Дан массив целых неотрицательных чисел a длины n и число k . За одну операцию можно изменить любой элемент массива на 1. Необходимо сделать так, чтобы разница всех соседних элементов массива a была не больше k по абсолютному значению, за минимальное количество операций. При этом делать элементы массива отрицательными запрещено.

Формат входных данных

В первой строке входных данных даны два целых неотрицательных числа n и k ($1 \leq n \leq 200\,000$, $0 \leq k \leq 10^9$) — длина массива a и максимальная допустимая разность соседних элементов.

Во второй строке входных данных даны n целых неотрицательных чисел a_i ($0 \leq a_i \leq 10^9$) — элементы массива a .

Формат выходных данных

Выведите одно число — минимальное количество операций, которые надо сделать, чтобы абсолютное значение разницы любых соседних элементов массива было не больше k .

Примеры

стандартный ввод	стандартный вывод
6 1 2 10 0 2 4 3	10
6 3 2 10 2 6 4 3	6
4 1 1 4 1 4	4
10 1 10 9 8 7 6 5 4 3 2 1	0
3 0 1 1 3	2