

Задача А. Калила и Димна на лесозаготовках

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

Спонсор сегодняшней задачи — codeforces round 189. Codeforces — мечты сбываются!

Калила и Димна — два шакала. Они живут в огромных джунглях. Однажды шакалы решили устроиться на завод лесозаготовки и подработать.

Управляющий завода хочет, чтобы они отправились в джунгли и срубили n деревьев высотой a_1, a_2, \dots, a_n . Для этого Калила и Димна купили цепную пилу в магазине. Каждый раз, когда они используют пилу на дереве номер i , они уменьшают высоту этого дерева на единицу. Каждый раз Калила и Димна должны заправить пилу для использования. Цена заправки зависит от того, какие деревья полностью спилены (дерево считается полностью спиленным, если его высота равна 0). Если максимальный идентификатор полностью срубленного дерева равняется i (первоначально это дерево имело высоту a_i), то цена заправки пилы равняется b_i . Если ни одно дерево не срублено полностью, то заправлять пилу запрещается. Изначально пила заправлена. Известно, что для каждого $i < j$, $a_i < a_j$ и $b_i > b_j$, а также $b_n = 0$ и $a_1 = 1$.

Калила и Димна хотят полностью срубить все деревья с минимальными затратами. Они ждут Вашей помощи! Поможете?

Формат входных данных

В первой строке записано целое число n ($1 \leq n \leq 10^5$). Во второй строке записано n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$). В третьей строке записано n целых чисел b_1, b_2, \dots, b_n ($0 \leq b_i \leq 10^9$).

Гарантируется, что $a_1 = 1$, $b_n = 0$, $a_1 < a_2 < \dots < a_n$ и $b_1 > b_2 > \dots > b_n$.

Формат выходных данных

В единственной строке должна быть записана минимальная стоимость вырубания всех деревьев.

Примеры

<code>lumber.in</code>	<code>lumber.out</code>
5 1 2 3 4 5 5 4 3 2 0	25
6 1 2 3 10 20 30 6 5 4 3 2 0	138

Задача В. Серверы

Имя входного файла: `server.in`
Имя выходного файла: `server.out`
Ограничение по времени: 2.5 секунд
Ограничение по памяти: 256 мегабайт

Компьютерная сеть в некотором доме строилась по принципу присоединения нового компьютера к последнему из уже подключенных. Никакие два компьютера, будучи подключенными в сеть, между собой дополнительно никак не связывались. Таким образом, в сеть были объединены последовательно N компьютеров. Соседи обменивались информацией между собой, но в какой-то момент поняли, что им нужны прокси-серверы. Компьютерное сообщество дома решило установить прокси-серверы ровно на K компьютеров. Осталось только решить, какие именно компьютеры выбрать для этой цели. Главным критерием является ежемесячная стоимость обслуживания серверами всех компьютеров.

Для каждого компьютера установлен тариф его обслуживания, выраженный в рублях за метр провода. Стоимость обслуживания одного компьютера каким-то сервером равна тарифу компьютера, умноженному на суммарную длину провода от этого компьютера до сервера, которым он обслуживается.

Ваша задача написать программу, которая выберет такие K компьютеров, чтобы установить на них прокси-серверы, что общие затраты на обслуживание всех компьютеров были бы минимальными.

Формат входных данных

В первой строке входного файла записано два целых числа N и K — количество компьютеров в сети и количество прокси-серверов, которые нужно установить ($1 \leq K \leq N \leq 2000$).

Все компьютеры в сети пронумерованы числами от 1 до N по порядку подключения.

Во второй строке записано одно целое число T_1 — тариф обслуживания первого компьютера.

В следующих $N - 1$ строках записано через пробел по два целых неотрицательных числа L_i , T_i — информация об остальных компьютерах в сети по порядку номеров. L_i — длина провода, соединяющего i -й компьютер с соседним с меньшим номером, T_i — тариф обслуживания данного компьютера ($2 \leq i \leq N$). Все L_i и T_i от 0 до 10^6 .

Формат выходных данных

В первую строку выходного файла необходимо вывести одно целое число — минимальную стоимость обслуживания всех компьютеров всеми серверами. Во второй строке должны быть записаны через пробел K номеров компьютеров, на которые необходимо установить серверы (номера разрешается выводить в любом порядке). При существовании нескольких вариантов размещения разрешается вывести любой.

Примеры

<code>server.in</code>	<code>server.out</code>
3 1 10 2 2 3 3	19 1
3 2 10 2 2 3 3	4 1 3

Задача С. Бэкапы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Вы управляете IT-компанией, которая создает бэкапы компьютерных данных для больших офисов. Делать бэкапы скучно, и поэтому вы разрабатываете свою систему таким образом, чтобы разные офисы могли создавать бэкапы данных друг друга, пока вы сидите дома и играете в компьютерные игры.

Все офисы ваших клиентов расположены вдоль одной улицы. Вы решили объединить офисы в пары, и для каждой пары офисов вы проложите сетевой кабель между двумя зданиями, чтобы они могли создавать бэкапы данных друг друга.

Однако сетевые кабели стоят дорого. Ваша местная телекоммуникационная компания предоставит вам только k сетевых кабелей, что означает, что вы можете организовать резервное копирование только для k пар офисов, (в общей сложности $2k$ офисов). Ни один офис не может принадлежать более чем одной паре (то есть все эти $2k$ офисов должны быть разными).

Кроме того, телекоммуникационная компания взимает плату за километр. Это означает, что вам нужно выбрать эти k пар офисов, чтобы использовать как можно меньше кабелей. Другими словами, вам нужно выбрать пары таким образом, чтобы при сложении расстояний между двумя офисами в каждой паре общее расстояние было как можно меньше.

Например, предположим, что у вас было пять клиентов, офисы которых расположены как показано ниже: в 1 км, 3 км, 4 км, 6 км и 12 км от начала улицы. Телекоммуникационная компания предоставит вам только $k = 2$ кабелей.

Лучше всего соединить первый и второй офис и третий и четвертый офис. Таким образом вы используете все $k = 2$ кабелей и длина первого кабеля будет $3 - 1 = 2$ км, и длина второго кабеля будет $6 - 4 = 2$ км. Таким образом, суммарная длина кабелей будет 4км. Можно показать, что меньшую длину получить нельзя.

Формат входных данных

Первая строка содержит два числа: n и k , обозначающих количество офисов на улице ($2 \leq n \leq 100\,000$) и количество доступных вам кабелей ($1 \leq k \leq \frac{n}{2}$).

Следующих n строк содержат по одному целому числу: ($0 \leq s \leq 1\,000\,000\,000$), i -ое число означает расстояние от i -ого офиса до начала улицы. Офисы отсортированы по расстоянию до начала улицы, от ближайшего до наиболее далёкого. Не существует двух офисов, которые находятся в одном месте.

Формат выходных данных

Вы должны вывести ровно одно целое положительное число - минимальную длину кабелей, необходимую чтобы соединить $2k$ офисов при помощи k кабелей.

Примеры

стандартный ввод	стандартный вывод
5 2 1 3 4 6 12	4

Замечание

Пример входных данных соответствует примеру разобранным в условии.

Задача D. Бэкапы с восстановлением

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Вы управляете IT-компанией, которая создает бэкапы компьютерных данных для больших офисов. Делать бэкапы скучно, и поэтому вы разрабатываете свою систему таким образом, чтобы разные офисы могли создавать бэкапы данных друг друга, пока вы сидите дома и играете в компьютерные игры.

Все офисы ваших клиентов расположены вдоль одной улицы. Вы решили объединить офисы в пары, и для каждой пары офисов вы проложите сетевой кабель между двумя зданиями, чтобы они могли создавать бэкапы данных друг друга.

Однако сетевые кабели стоят дорого. Ваша местная телекоммуникационная компания предоставит вам только k сетевых кабелей, что означает, что вы можете организовать резервное копирование только для k пар офисов, (в общей сложности $2k$ офисов). Ни один офис не может принадлежать более чем одной паре (то есть все эти $2k$ офисов должны быть разными).

Кроме того, телекоммуникационная компания взимает плату за километр. Это означает, что вам нужно выбрать эти k пар офисов, чтобы использовать как можно меньше кабелей. Другими словами, вам нужно выбрать пары таким образом, чтобы при сложении расстояний между двумя офисами в каждой паре общее расстояние было как можно меньше.

Например, предположим, что у вас было пять клиентов, офисы которых расположены как показано ниже: в 1 км, 3 км, 4 км, 6 км и 12 км от начала улицы. Телекоммуникационная компания предоставит вам только $k = 2$ кабелей.

Лучше всего соединить первый и второй офис и третий и четвертый офис. Таким образом вы используете все $k = 2$ кабелей и длина первого кабеля будет $3 - 1 = 2$ км, и длина второго кабеля будет $6 - 4 = 2$ км. Таким образом, суммарная длина кабелей будет 4км. Можно показать, что меньшую длину получить нельзя.

Формат входных данных

Первая строка содержит два числа: n и k , обозначающих количество офисов на улице ($2 \leq n \leq 100\,000$) и количество доступных вам кабелей ($1 \leq k \leq \frac{n}{2}$).

Следующих n строк содержат по одному целому числу: ($0 \leq s \leq 1\,000\,000\,000$), i -ое число означает расстояние от i -ого офиса до начала улицы. Офисы отсортированы по расстоянию до начала улицы, от ближайшего до наиболее далёкого. Не существует двух офисов, которые находятся в одном месте.

Формат выходных данных

Вы должны сначала вывести одно целое положительное число - минимальную длину кабелей, необходимую чтобы соединить $2k$ офисов при помощи k кабелей.

Затем нужно вывести k строк, в каждой по два числа от 1 до N — номера домов, которые нужно соединить кабелем. Вы можете выводить пары в любом порядке.

Примеры

стандартный ввод	стандартный вывод
5 2	4
1	3 4
3	1 2
4	
6	
12	

Замечание

Пример входных данных соответствует примеру разобранным в условии.

Задача Е. Путешествие

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Маленький Петя очень любит путешествовать. В стране Берляндия, где он живет, есть n городов, расположенных на одной прямой. Петя пронумеровал их числами от 1 до n в порядке увеличения красоты. Петя находится в городе 1 и хочет попасть в город n . Чтобы не портить впечатления о поездке, он может посещать города только в порядке увеличения номеров (а, следовательно, и красоты).

Для перемещения между городами Петя решил воспользоваться услугами единственной авиакомпании страны — Berland Airlines. Стоимость перелёта из города i в город j равна $c_i \cdot |x_i - x_j| + t_j$, где x_i — координата города i , x_j — координата города j , а c_i — стоимость единицы самолётного топлива в городе i , а t_j — стоимость въезда в город j .

Чтобы было о чем рассказать друзьям, Петя хочет потратить как можно больше (да-да, именно больше) денег на эту поездку. Помогите ему в этом. Обратите внимание, что Пете не обязательно бывать во всех городах.

Формат входных данных

Первая строка содержит целое число n — количество городов в Берляндии ($1 \leq n \leq 100\,000$).

Далее следуют n строк. Строка с номером i содержит три целых числа — x_i , c_i и t_i ($-10^6 \leq x_i \leq 10^6$, $1 \leq c_i \leq 10^6$, $1 \leq t_i \leq 10^6$).

Формат выходных данных

Выведите искомое наибольшее количество денег, которые Петя может потратить чтобы добраться из города 1 в город n . Гарантируется, что ответ не превосходит 10^{12} .

Примеры

стандартный ввод	стандартный вывод
4	123
5 10 2	
0 1 10	
15 3 14	
17 2 3	

Задача F. Оптимальное бинарное дерево поиска

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

Рассмотрим множество $S = \{e_1, e_2, \dots, e_n\}$, состоящее из n различных элементов таких, что $e_1 < e_2 < \dots < e_n$. Рассмотрим бинарное дерево поиска, состоящее из элементов S . Чем чаще производится запрос к элементу, тем ближе он должен располагаться к корню. Стоимостью $cost$ доступа к элементу e_i из S в дереве будем называть значение $cost(e_i)$, равное числу ребер на пути, который соединяет корень с вершиной, содержащей элемент. Имея частоту запросов к элементам из S — $f(e_1), f(e_2), \dots, f(e_n)$ — определим общую стоимость дерева следующим образом:

$$\sum_{i=1}^n f(e_i) \cdot cost(e_i)$$

Дерево, имеющее наименьшую стоимость, считается наилучшим для поиска элементов из S . Именно поэтому оно называется Оптимальным Бинарным Деревом Поиска. Ваша задача — найти стоимость Оптимального Бинарного Дерева Поиска.

Формат входных данных

Состоит из нескольких тестов, каждый из которых расположен в отдельной строке. Первое число в строке n ($1 \leq n \leq 5000$) указывает на размер множества S . Следующие n неотрицательных целых чисел описывают частоты запросов элементов из S : $f(e_1), f(e_2), \dots, f(e_n)$. Известно, что $0 \leq f(e_i) \leq 100$. Сумма n по всем тестам не больше 5000.

Формат выходных данных

Для каждого теста в отдельной строке выведите стоимость Оптимального Бинарного Дерева Поиска.

Пример

стандартный ввод	стандартный вывод
1 5	0
3 10 10 10	20
3 5 10 20	20
6 1 3 5 10 20 30	63

Задача G. Гонки в линейном королевстве

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

Вы — организатор гонок, и вы хотите провести серию заездов в Линейном Королевстве.

В Линейном Королевстве n дорог, пронумерованных от 1 до n . Дороги идут одна за другой слева направо, в порядке возрастания номера. Вам поступило несколько предложений проведения заездов. Для каждого заезда известно, на каких дорогах предлагается его проводить — некоторый набор последовательных дорог Королевства. Также известно ваше вознаграждение, в случае если вы решите провести данный заезд. Все заезды предлагается проводить в разное время, то есть одна и та же дорога может использоваться в нескольких заездах.

К несчастью, некоторые дороги в плохом состоянии, и их нужно чинить. Для каждой дороги известна стоимость ее ремонта. Заезд можно проводить только если отремонтированы все дороги, на которых проходит гонка. Ваша задача — отремонтировать такой набор дорог (возможно, все, или ни одной), чтобы ваша прибыль была как можно больше. Прибыль считается как сумма вознаграждений, полученных за проведение заездов, минус сумма денег, потраченных на ремонт дорог. Заметьте, что вы можете не чинить дороги вообще и получить нулевую прибыль.

Найдите максимальную возможную прибыль.

Формат входных данных

В первой строке через пробел записано два целых числа n и m ($1 \leq n, m \leq 2 \cdot 10^5$) — количество дорог и предложенных заездов соответственно.

Далее следует n строк. Каждая строка содержит одно целое неотрицательное число, не превосходящее 10^9 — стоимость починки очередной дороги. Стоимости починки даны в порядке начиная с дороги номер 1 заканчивая дорогой номер n .

Далее следует m строк по три числа в каждой: lb , ub , и p ($1 \leq lb \leq ub \leq n, 1 \leq p \leq 10^9$). Каждая строка описывает очередной заезд: предлагается использовать дороги с lb по ub включительно, и вознаграждение будет составлять p .

Формат выходных данных

Выведите одно целое число — максимальную возможную прибыль, которую вы можете получить.

Примеры

стандартный ввод	стандартный вывод
7 4 3 2 3 2 1 2 3 1 2 5 2 3 5 3 5 3 7 7 5	4
2 1 0 3 1 2 5	2
3 1 10 10 10 1 3 10	0

Замечание

В первом примере выгодно чинить дороги 1, 2, 3 и 7. Три заезда приносят доход 15. Ремонт стоит 11, значит прибыль равна 4.

Задача Н. А ты сортируешь запросы?

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 1024 мегабайта

Благодарим DittuT за предоставленную задачу.

Вам даётся N запросов, которые бывают трёх типов:

1. Добавить пару чисел (a, b) в набор. $(-10^9 \leq a, b \leq 10^9)$
2. Удалить пару добавленную в запросе $index$. (Запросы нумеруются числами от 1 до N в порядке в котором они идут вводятся в программу.
3. Вам даётся число A и требуется найти максимум $a_i \cdot A + b_i$ по всем парам в наборе. $(-10^9 \leq A \leq 10^9)$. Гарантируется, что в наборе есть хоть одна пара в этот момент.

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 3 \cdot 10^5$) - количество запросов.

Каждая из следующих n строк начинается с целого числа op ($1 \leq op \leq 3$) - типа запроса. Далее, в зависимости от типа запроса, следуют, либо числа a_i и b_i , либо число $index$, либо число A .

Формат выходных данных

Для каждого запроса типа 3 выведите в отдельной строке максимальное значение $a \cdot A + b$. Гарантируется, что в наборе есть хоть одна пара в этот момент.

Примеры

стандартный ввод	стандартный вывод
4	-1000000000000000000
1 -1000000000 0	-999999999999999999
3 1000000000	
1 -1000000000 1	
3 1000000000	

Задача I. Минимальный разрез

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Мощностью массива $[b_1, \dots, b_m]$ назовём величину $M([b_1, \dots, b_m]) = b_1 \cdot 1 + b_2 \cdot 2 + \dots + b_m \cdot m$.

Вам дан массив $[a_1, \dots, a_n]$, стоимость разреза между индексами i и $i + 1$ равна c_i . Найдите такое разбиение массива на части, чтобы минимизировать сумму мощностей частей и стоимостей разрезов.

Формально, найдите такие целые индексы $1 \leq i_1 < \dots < i_k < n$, что

$$M([a_1, \dots, a_{i_1}]) + c_{i_1} + M([a_{i_1+1}, \dots, a_{i_2}]) + c_{i_2} + \dots + c_{i_k} + M([a_{i_k+1}, \dots, a_n])$$

минимально.

Формат входных данных

В первой строке записано единственное целое число n ($2 \leq n \leq 10^6$) — длина данного массива.

Во второй строке записано n целых чисел a_1, \dots, a_n — сам массив ($0 \leq a_i \leq 10^6$).

В третьей строке записаны $n - 1$ целое число c_1, \dots, c_{n-1} — стоимости разрезов ($0 \leq c_i \leq 10^{12}$).

Формат выходных данных

Выведите одно целое число — минимальную сумму мощностей частей и стоимостей разрезов.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3 9 8	14
4 1 2 3 5 3 2 7	20

Замечание

В первом примере массив выгодно вообще не разбивать.

Во втором примере выгодно массив разбить на две части по два элемента. Тогда $M([1, 2]) + 2 + M([3, 5]) = 5 + 2 + 13 = 20$.