

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

lumber.in	lumber.out
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. Сiel и гондолы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Лиса Сiel зашла в парк аттракционов. И вот, она в очереди на колесо обозрения. В очереди стоит n людей (хотя нет, скорее лис): мы будем считать, что первая лиса стоит в начале очереди, а n -я лиса стоит в хвосте очереди.

Всего имеется k гондол, мы распределяем лис по гондолам следующим образом:

- Когда подплывает первая гондола, q_1 лис переходят из начала очереди в подплывшую гондолу.
- Затем, когда подплывает вторая гондола, q_2 лис из начала оставшейся очереди переходит в эту гондолу.
- ...
- Оставшиеся q_k лис идут с последней (k -ю) гондолу.

Обратите внимание, что числа q_1, q_2, \dots, q_k должны быть положительными. Из условия следует, что $\sum_{i=1}^k q_i = n$ и $q_i > 0$.

Вы знаете как лисам не хочется задерживаться в гондолах с незнакомцами. Итак, Ваша задача — найти оптимальный способ размещения (то есть определить оптимальную последовательность q), чтобы угодить всем. Для каждой пары лис i и j задано значение u_{ij} , обозначающее степень незнакомости. Можете считать, что $u_{ij} = u_{ji}$ для всех i, j ($1 \leq i, j \leq n$) и что $u_{ii} = 0$ для всех i ($1 \leq i \leq n$). Тогда значение незнакомости в гондоле определяется как сумма значений незнакомости между всеми парами лис, которые находятся в этой гондоле. Общее значение незнакомости определяется как сумма значений незнакомости по всем гондолам.

Помогите лисе Сiel найти минимальное возможное значение общей незнакомости при некотором оптимальном распределении лис по гондолам.

Формат входных данных

В первой строке даны два целых числа n и k ($1 \leq n \leq 4000$ and $1 \leq k \leq \min(n, 800)$) — количество лис в очереди и количество гондол. В следующих n строках записано по n целых чисел — матрица u , ($0 \leq u_{ij} \leq 9$, $u_{ij} = u_{ji}$ и $u_{ii} = 0$).

Пожалуйста, используйте методы быстрого чтения (например, для Java используйте `BufferedReader` вместо `Scanner`).

Формат выходных данных

Выведите целое число — минимальное возможное значение общей незнакомости.

Примеры

стандартный ввод	стандартный вывод
5 2 0 0 1 1 1 0 0 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0	0
8 3 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0	7

Замечание

В первом примере можно распределить лис вот так: 1, 2 идут в одну гондолу, 3, 4, 5 идут в другую гондолу.

Во втором примере оптимальное распределение таково: 1, 2, 3 | 4, 5, 6 | 7, 8.

Задача С. 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	19 1
2 2 3 3	
3 2 10	4 1 3
2 2 3 3	

Задача D. 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	4
1	
3	
4	
6	
12	

Замечание

Пример входных данных соответствует примеру разобранным в условии.

Задача Е. 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 F. 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 G. Гонцы

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

Давным-давно в прекрасном молдавском государстве существовало N средневековых городов, пронумерованных от 1 до N . Город с номером 1 был столицей. Города были соединены $N - 1$ двусторонними дорогами. Каждая дорога имела свою длину. Существовал единственный способ проехать между любыми двумя городами (т.е. дороги образовывали дерево).

Когда город подвергался нападению, о ситуации нужно было как можно скорее сообщать в столицу. Послания передавались при помощи гонцов. В каждом городе жил ровно один гонец. Каждый гонец характеризовался временем, необходимым ему для начала путешествия, и постоянной скоростью, с которой он двигался после отбытия.

Сообщение всегда передавалось по уникальному пути от города до столицы. Когда гонец приезжал в город, у него было два варианта дальнейших действий: вести сообщение дальше самому, или передать его другому гонцу. Новый гонец тоже тратил некоторое время, а потом ехал в столицу с постоянной скоростью. Одно сообщение могло быть передано при помощи любого количества гонцов.

Вам необходимо для каждого города узнать за какое минимальное время можно передать сообщение из него в столицу.

Формат входных данных

Первая строка содержит одно целое число N , которое означает количество городов в государстве ($3 \leq N \leq 100\,000$). Каждая из следующих $N - 1$ строк содержит три целых числа u , v и d , описывающих дорогу длиной d километров между городами, с номерами u и v ($1 \leq u, v \leq n$, $0 \leq d \leq 10\,000$).

Затем следуют $N - 1$ пар целых чисел, по одной паре на строку. Пара $i, S_i V_i$, описывает характеристики гонца в $(i + 1)$ -ом городе: S_i - это количество минут для подготовки к путешествию, а V_i - это количество минут, необходимое для прохождения одного километра ($0 \leq S_i \leq 10^9$, $1 \leq V_i \leq 10^9$). В столице нет гонца.

Формат выходных данных

Выходные данные должны состоять ровно из $N - 1$ целых чисел. Число i -это минимальное время в минутах, необходимое для отправки сообщения из города $(i + 1)$ в столицу.

Примеры

стандартный ввод	стандартный вывод
5	206 321 542 328
1 2 20	
2 3 12	
2 4 1	
4 5 3	
26 9	
1 10	
500 2	
2 30	

Задача Н. Задача без легенды

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

Дан неубывающий массив a размера n , а также стоимостей элементов c . Вам нужно выбрать подпоследовательность (возможно, пустую) с минимальной стоимостью и вывести эту стоимость. Стоимость подпоследовательности из элементов i_1, i_2, \dots, i_k это сумма стоимостей ее элементов плюс сумма «расстояний» между ее соседними элементами. «Расстояние» между элементами i и j , ($i < j$) — это $(7a_j - 4a_i)(5a_j - 2a_i)(4a_j - a_i)$.

Формат входных данных

В первой строке вводится натуральное число n ($1 \leq n \leq 10^6$). Во второй строке вводится массив целых чисел a ($0 \leq a_i \leq 10^6$). В третьей строке вводится массив целых чисел c ($-10^{18} \leq c_i \leq 10^{18}$)

Формат выходных данных

Выведите одно число — минимальную стоимость подпоследовательности.

Примеры

стандартный ввод	стандартный вывод
2 0 1 2 -6	-6
3 1 2 4 -6 5 1	-6
5 1 2 2 2 4 0 -2 -1 -5 2	-5
4 1 1 3 5 7 -10 8 -7	-10
5 0 1 4 4 4 -8 2 10 4 5	-8
2 1 1 -100 -100	-173

Замечание

Так как ответ может быть очень большим, рекомендуется использовать `__int128` или еще что-нибудь.