Задача A. Поймать Халка

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

Когда Локи ловил Халка, он немного не рассчитал своих сил, и случайно перенес его в параллельный n-мерный мир. После этого Локи намертво вморозил Халка в глыбу льда. Для окончательной победы Локи необходимо только отпилить от глыбы лишний лед так, чтобы остался только сам замороженный Халк.

Пространство, в которое Локи перенес все происходящее, не более чем трехмерно. В одномерном пространстве глыба представляет из себя отрезок некоторой длины, а Халк внутри — вложенный в него отрезок. В двумерном пространстве глыба и Халк — прямоугольники со сторонами, параллельными осям координат, причем Халк вложен в глыбу. Аналогично, в трехмерном пространстве глыба и Халк являются параллелепипедами со сторонами, параллельными осям координат.

Локи может отрезать от глыбы какие-то куски льда. В одномерном пространстве разрез — точка, в двумерном — прямая, в трехмерном — плоскость. В любом пространстве разрез не должен проходить через Халка, но может его касаться. Локи хочет узнать, за какое минимальное количество разрезов он сможет оставить от глыбы льда только ту ее часть, в которой находится Халк.

Формат входного файла

Первая строка входного файла содержит одно число n ($1 \le n \le 3$) — количество измерений в пространстве, в котором происходит действие.

Следующая строка содержит n натуральных чисел a_i ($1 \le a_i \le 10000$) — координаты одной из вершин глыбы. Будем считать, что вершина глыбы, противоположная данной, находится в начале координат.

В следующей строке сначала перечислены n целых чисел b_i ($0 \le b_i \le a_i$) — координаты одной из вешин Халка, затем еще n целых чисел c_i ($0 \le c_i \le a_i$) — координаты противоположной вершины Халка.

Формат выходного файла

Выведите единственное целое число — минимальное количество разрезов, которые необходимо сделать Локи, чтобы выпилить Халка.

Примеры

frozen.in	frozen.out
1 5 0 3	1
2 3 4 2 2 3 3	3
3 2 2 2 0 1 0 1 2 1	3

Комментарий

Решения, работающие в случаях, в которых $n = 1$, будут оцениваться в 20 баллов.

Решения, работающие в случаях, в которых $n \le 2$, будут оцениваться в 60 баллов.

Задача B. Держать строй

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

Ник Фьюри решил, что бойцы отряда спецназа, являющегося подразделением организации S.H.I.E.L.D., помогут мстителям отразить атаку войска Локи. Он решил, что в бой отправятся n бойцов, а все остальные понадобятся в других местах. Теперь ему осталось только выбрать, какие именно бойцы пойдут в атаку.

Сначала Ник выбрал n бойцов случайным образом и выстроил их в линию, а затем стал по одному заменять кого-то из уже выбранных бойцов на другого солдата, который в настоящее время в строю не стоит. Поскольку отряд достаточно большой, Ник не знает каждого бойца лично. Оценить боеспособность отряда он может разве что по каким-нибудь заметным внешним признакам. Важным показателем боеспособности отряда является, например, то, стоят ли солдаты в строю по неубыванию роста.

Так, Ник может давать команды двух видов. Первая команда заключается в том, что новый солдат роста x встает в строй вместо солдата, стоящего на k-ом месте. Подавая вторую команду, он хочет узнать, стоят ли солдаты в строю по неубыванию роста. Ваша задача обрабатывать эти команды и сообщать в ответ на запросы то, что хочет узнать Ник.

Формат входного файла

Первая строка входного файла содержит два числа n и m ($1 \le n \le 100\,000$, $0 \le m \le 200\,000$) — количество солдат в строю и количество команд, которые подаст Ник. Вторая строка содержит n целых неотрицательных чисел, не превосходящих 10^9 — исходный рост солдат в строю.

Следующие m строк содержат команды, подаваемые Ником. Если первый символ в строке, описывающей очередную команду, ‘!’, то за ним следуют два числа k и x ($1 \le k \le n$, $0 \le x \le 10^9$), где k — место в строю того солдата, которого должен заменить солдат роста x. Команда второго типа описывается знаком ‘?’.

Формат выходного файла

Для каждой команды второго типа в отдельной строке выведите «YES», если в данный момент солдаты в строю стоят по неубыванию роста, и «NO» — в противном случае.

Примеры

army.in	army.out
5 5 2 4 6 8 10 ? ! 2 7 ? ! 3 8 ?	YES NO YES

Комментарий

Решения, работающие в случаях, в которых n и m не превышает 10 000, будут оцениваться в 60 баллов.

Задача C. Тессеракт

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

Коварный Локи напал на Землю, использовав Тессеракт для того, чтобы переместиться из Асгарда. Понимая ценность этого артефакта, он решает защитить его от использования кем-либо, кроме себя. Для этого он решает немного изменить его внутреннюю структуру так, чтобы только он знал, как её можно восстановить.

После внимательного исследования, внимание Локи привлекла цепочка маленьких кубиков внутри Тессеракта. Среди этих кубиков были крайне похожие друг на друга, что идеально подходило для небольшого, но важного для работы изменения Тессеракта. Проявив восхвалённую в легендах коварность, Локи, в надежде на невнимательность людей, решил внести два изменения таких, чтобы после каждого из них, артефакт выглядел бы таким же, как и раньше.

Всего цепочка, которую собирается менять Локи, состоит из n кубиков. Среди них есть похожие, на чём и собирается сыграть Локи. Он берёт какой-то отрезок кубиков в этой цепочке, вырывает его, разворачивает и вставляет обратно. При этом, он выбирает отрезок таким образом, чтобы внешне цепочка не изменилась. Затем он повторяет то же самое с другим отрезком, который содержал в себе первый, но не совпадал с ним. Очевидно, после этих двух операций цепочка уже не будет совпадать с исходной, и артефакт будет испорчен.

Чтобы оценить вероятность быть уличённым в порче Тессеракта, Локи решил выяснить, сколькими способами он мог выбрать первый отрезок.

Рассмотрим, к примеру, цепочку $aabaa$, в которой одинаковыми буквами обозначены похожие кубики. Тогда настоящий Тессеракт содержит цепочку $a_1a_2ba_3a_4$. Локи может, например, проделать следующую последовательность действий: $a_1a_2ba_3a_4 \to a_2a_1ba_3a_4 \to a_4a_3ba_1a_2$. Внешне ничего не изменилось, однако цепочка уже другая.

Формат входного файла

В первой и единственной строке задана цепочка, состоящая из маленьких латинских букв, длиной не более 100 000.

Формат выходного файла

Единственное число — количество различных первых действий Локи.

Примеры

tesseract.in	tesseract.out
aabaa	8

Комментарий

Решения, работающие для строк длины не более 100, будут оцениваться в 30 баллов.

Решения, работающие для строк длины не более 4000, будут оцениваться в 60 баллов.

Задача D. Эвакуация

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

Через T минут армия читаури под предводительством Локи атакует Землю. Мстители никак не успевают помешать открытию портала в Нью-Йорке, поэтому Капитан Америка принял решение эвакуировать из города всех его жителей. Ему необходимо выяснить, успеют ли жители города эвакуироваться до начала вторжения.

Окрестности Нью-Йорка можно представить как набор небольших городов, связанных между собой дорогами с односторонним движением. Каждая дорога характеризуется своей длиной и пропускной способностью. Длина дороги l означает, что въехав на нее в момент времени t, автомобиль окажется в конце этой дороги через l минут, в момент времени $t + l$. Пропускная способность дороги s означает, что каждую минуту на эту дорогу могут въехать не больше, чем s автомобилей. Приехав в какой-нибудь город, любой автомобиль может сразу продолжить путь, въехав на какую-то дорогу, выходящую из этого города, а может остановиться в этом городе на любое количество минут, и только потом уехать из него.

Капитан Америка уже решил, в каком именно городе должны оказаться жители Нью-Йорка после эвакуации. Также ему известно, сколько автомобилей необходимо для эвакуации всего города. Теперь ему необходимо выяснить успеют ли все жители эвакуироваться до прибытия захватчиков и, если да, какое минимальное количество времени у них на это уйдет, а если нет — какому минимальному числу автомобилей с горожанами не удастся доехать до безопасного города.

Формат входного файла

Первая строка входного файла содержит четыре целых числа n, m, K и T ($1 \le n \times T \le 10\,000$, $1 \le m, K \le 10\,000$) — количество городов в окрестностях Нью-Йорка, количество дорог между ними, количество автомобилей, которым необходимо попасть из Нью-Йорка в безопасный город и время до вторжения захватчиков соответственно. Следующие m строк содержат описания дорог между городами.

Каждая дорога описывается четырьмя целыми числами u, v, l и s ($1 \le u, v \le n$, $u \ne v$, $1 \le s \le 3\,000$, $1 \le l \le 200$) — город, из которого выходит эта дорога, город, в который она ведет, ее длина и пропускная способность соответственно.

Между двумя городами может существовать только одна дорога, ведущая в каком-то направлении. Нью-Йорком считается город с номером 1, а безопасным городом — город с номером n. в момент времени 0 все автомобили находятся в Нью-Йорке.

Формат выходного файла

Если все жители Нью-Йорка успеют добраться до безопасного города не более, чем за T минут, выведите в выходной файл минимальное количество минут, которое им на это понадобится. В противном случае выведите минимальное количество автомобилей, которым не удастся попасть в безопасное место за T минут. Да, не нужно выводить, какой из этих случаев имеет место :-).

Примеры

evacuation.in	evacuation.out
5 5 10 10 1 2 2 2 2 3 1 1 2 4 1 1 4 5 2 4 3 5 2 4	9

Комментарий

Решения, корректно работающие при $n \times T \leq 500$, будут оцениваться в 60 баллов.