

Задача А. Красивый массив

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Вам дан массив a состоящий из n целых чисел. Красота массива – это максимальная сумма какого-то последовательного подотрезка этого массива (этот подотрезок может быть пустым). Например красота массива $[10, -5, 10, -4, 1]$ равна 15, а красота массива $[-3, -5, -1]$ равна 0.

Вы можете выбрать не более одного последовательного подотрезка массива a и домножить все элементы этого подотрезка на x . Вы хотите максимизировать красоту массива после применения такой операции.

Формат входных данных

Первая строка содержит два целых числа n и x , ($1 \leq n \leq 3 \cdot 10^5, -100 \leq x \leq 100$) — длина массива a и число x соответственно.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n , ($-10^9 \leq a_i \leq 10^9$) — сам массив a .

Формат выходных данных

Выведите одно число — максимальную красоту массива a после не более одного домножения непрерывного подотрезка этого массива на x .

Примеры

стандартный ввод	стандартный вывод
5 -2 -3 8 -2 1 -6	22
12 -3 1 3 3 7 1 3 3 7 1 3 3 7	42
5 10 -1 -2 -3 -4 -5	0

Задача В. Диаметр дерева

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

Вам дано дерево из n вершин. Требуется вывести максимальную длину простого пути в данном дереве.

Формат входных данных

Первая строка каждого теста содержит натуральное число n — количество вершин в дереве ($1 \leq n \leq 100\,000$). Следующие $n - 1$ строк содержат по 2 натуральных числа и описывают ребра дерева.

Формат выходных данных

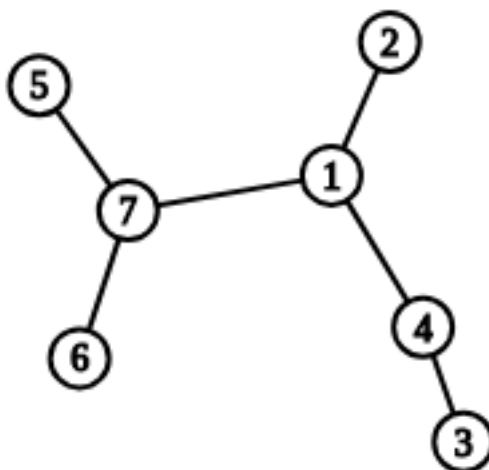
Выведите единственное число — максимальную длину простого пути в дереве.

Пример

стандартный ввод	стандартный вывод
7 1 2 3 4 7 6 7 5 7 1 1 4	4

Замечание

Пояснения к первому примеру:



Задача С. Дерзкие ограбления

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

В стране Олимпия очень развита банковская система, поэтому жители охотно кладут свои сбережения в банк. Всего есть n банков, которые стоят в ряд. В банке с номером i лежат a_i тугриков. Изначально в банках нет системы безопасности, которая могла бы помешать ограблению. Однако известно, что если вечером дня с номером d ограбили банк с номером b , то на утро следующего дня необходимая система безопасности будет установлена в соседних банках ($b - 1$ и $b + 1$), после чего их уже невозможно будет ограбить. Утром дня с номером $d + i$ ($i > 0$) система безопасности будет установлена в банках $b - i$ и $b + i$. Так будет продолжаться, пока все банки не будут защищены от ограбления. Ограбленный банк уже никогда нельзя ограбить.

К сожалению, в Олимпии даже преступники занимаются решением олимпиадных задач, поэтому правительство не сомневается в том, что если кто-то задумает провести серию ограблений, то он сделает это оптимальным образом, иначе говоря – максимизирует суммарное количество тугриков в тех банках, которые ему удастся ограбить до того, как в них будет установлена система безопасности. Кроме того, известно, что преступники грабят не более чем один банк в день, а также то, что они орудуют вечером. Банковская система, анализируя возможные убытки от ограблений, рассматривает последовательно $m + 1$ вариант размещения средств в банках. Каждый вариант отличается от предыдущего количеством денег в одном из банков.

Формат входных данных

В первой строке входного файла содержатся числа n и m – количество банков и операций изменения количества тугриков ($1 \leq n, m \leq 100\,000$). В следующей строке записано n чисел a_i , задающие начальное количество тугриков в банках. Далее следуют m строк, в каждой из которых записана пара чисел b и t , что означает, что после очередной операции в банке номер b станет t тугриков.

Формат выходных данных

В выходной файл для каждого i ($0 \leq i \leq m$) в отдельную строку выведите максимальное количество тугриков, которую смогут украсть преступники после выполнения i -й операций.

Пример

стандартный ввод	стандартный вывод
7 4	17
6 7 5 6 2 2 4	18
6 5	18
7 2	19
7 6	19
4 6	

Задача D. Шоколадка

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

64 мегабайта

Команда «Отбой» участвует в очередном марафоне по «Угадай мелодию. Rock version». Чтобы было чем подкрепиться во время игры, команда взяла с собой большую прямоугольную плитку шоколада размерами $w \times h$. У команды есть список из n пар чисел — размеры шоколадок, которые команда считает счастливыми. Прежде чем приступить к поеданию шоколадки, участники команды решили поделить имеющуюся плитку на счастливые шоколадки. Для этого они действуют следующим образом: сначала плитка шоколада ломается на 2 части по линии, строго параллельной одной из своих сторон, после чего каждую из полученных частей они могут продолжить ломать аналогичным образом.

Вам поручили определить, какое максимальное количество счастливых шоколадок команда сможет получить, действуя по данной схеме. Шоколадки, полученные поворотом счастливых, счастливыми не являются.

Формат входных данных

В первой строке входного файла заданы три целых числа w, h, n — размеры плитки шоколада и количество вариантов размера счастливых шоколадок соответственно ($1 \leq w, h \leq 300, 1 \leq n \leq w \times h$). В следующих n строках заданы пары целых чисел w_i, h_i — размеры счастливых шоколадок ($1 \leq w_i \leq w, 1 \leq h_i \leq h$).

Формат выходных данных

В единственную строку выходного файла выведите максимальное количество счастливых шоколадок, на которые можно разрезать данную плитку.

Примеры

стандартный ввод	стандартный вывод
21 11 4 10 4 6 2 7 5 15 10	15
9 12 5 1 12 2 6 3 4 4 3 6 2	9

Задача Е. Покраска забора

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.7 секунд
Ограничение по памяти: 256 мегабайт

256 мегабайт

Мэр города Многоярославца решил построить перед своим домом забор из n деревянных досок и нанять лучшего маляра города для его покраски. Поскольку забор должен стать главной достопримечательностью города, лучший дизайнер города для каждой доски назначил тщательно выбранный цвет, в который она должна быть покрашена.

Для покраски главный маляр решил применить новейшую технологию, специально разработанную им для выполнения этого задания. Покраской забора будет заниматься специальный робот, который за один час может покрасить произвольный отрезок забора (набор соседних досок) в некоторый цвет. Поскольку задание должно быть выполнено как можно быстрее, требуется составить программу для робота, которая позволит достичь требуемой раскраски за минимальное время. Оставить какую-то из досок непокрашенной, естественно, запрещается.

Формат входных данных

В первой строке входного файла записано число n ($1 \leq n \leq 300$), где n количество досок в заборе. Вторая строка содержит строку из n символов, описывающую требуемую покраску забора. Цвета обозначаются заглавными латинскими буквами.

Формат выходных данных

В первой строке выходного файла выведите m - наименьшее возможное время покраски забора в часах. Следующие m строк должны содержать программу покраски для робота. Каждая строка должна содержать два числа l_i и r_i , а также заглавную букву латинского алфавита, задающую цвет c_i и означает, что робот должен покрасить участок забора с l_i по r_i - доску в цвет c_i (если длина забора n , должно выполняться $1 \leq l_i \leq r_i \leq n$).

Примеры

стандартный ввод	стандартный вывод
5 АВВСА	3 1 5 А 2 3 В 4 4 С
2 АА	1 1 2 А

Задача F. Максимальное подмножество вершин

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

Надо бы всё-таки написать нормальную легенду, а то как-то не очень. И без легенды непонятно, почему задача так называется

Но пока легенды нет, вот формальное условие:

Вам дано дерево на n вершинах. В вершинах записаны числа a_n .

Требуется выбрать подмножество вершин с максимальной суммой a_n , чтобы никакие две соседние вершины не лежали одновременно в этом подмножестве.

Формат входных данных

В первой строке записано число n — количество вершин дерева ($1 \leq n \leq 10^6$). Во второй строке записаны через пробел n чисел v_i ($|v_i| < 10^9$), задающие значения в вершинах. В следующих $n - 1$ строках описаны ребра дерева. В $(i + 2)$ -й строке записаны номера вершин a_i, b_i ($1 \leq a_i, b_i \leq n$), означающие, что в дереве есть ребро из вершины a_i в вершину b_i .

Формат выходных данных

Выведите единственное число — максимальную искомую величину.

Пример

стандартный ввод	стандартный вывод
7 0 7 2 7 1 4 8 3 6 7 6 1 3 5 1 4 6 2 1	25

Задача G. Сумма длин путей 2

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

Дано дерево на n вершинах. На каждом ребре написан его вес. Требуется для каждого v посчитать сумму взвешенных длин всех путей в данном дереве, исходящих из v . Пути $\langle v, u \rangle$ и $\langle u, v \rangle$ считаются различными.

Формат входных данных

Первая строка каждого теста содержит натуральное число n — количество вершин в дереве ($1 \leq n \leq 100\,000$). Следующие $n - 1$ строк содержат по 3 натуральных числа v, u, w и описывают ребро дерева, соединяющее две вершины v и u и имеющее вес w ($1 \leq v, u \leq n, 0 \leq w \leq 10^6$).

Формат выходных данных

Выведите n чисел — требуемое в условии.

Примеры

стандартный ввод	стандартный вывод
3 1 2 1 1 3 3	4 5 7
7 1 2 1 3 4 1 7 6 1 7 5 1 7 1 1 1 4 1	9 14 17 12 15 15 10

Задача Н. Автостанции

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

Фирма, в которой работает ваш друг, решила воспользоваться удобным моментом и купила компанию, занимающуюся пригородными автобусными пассажирскими перевозками. Таким образом, фирма вашего друга расширяет область деятельности и будет теперь обслуживать и некоторые внутриобластные автобусные маршруты.

Сейчас руководство фирмы, и в том числе ваш друг, заняты оптимизацией работы этих маршрутов. Одна из основных проблем, которые были обнаружены, состоит в том, что большинство автобусов, использующихся там, очень старые и изношенные, и поэтому часто выходят из строя. В целях улучшения ситуации было принято решение о создании сети ремонтных подстанций, которые будут располагаться в некоторых населённых пунктах области и обслуживать другие близлежащие населённые пункты.

Система дорог в области устроена следующим простым образом. Есть N населённых пунктов, некоторые из которых соединены дорогами. Между каждой парой пунктов существует не более одной дороги, и более того, для каждой пары населённых пунктов есть ровно один способ добраться из одного в другой (возможно, через промежуточные посёлки).

В каждом населённом пункте можно разместить ремонтную подстанцию. В принципе, фирма может размещать как крупные подстанции, которые даже в одиночку смогут обслуживать всю область, но при этом будут требовать больших расходов на содержание, так и небольшие станции, которые будут обслуживать лишь прилегающие населённые пункты, но при этом будут обходиться намного дешевле. Фирма уже определила, что каждую подстанцию можно характеризовать параметром «мощность», которая может принимать значения, являющиеся целыми положительными числами (равна нулю мощность быть не может). Подстанция с мощностью k будет обслуживать населённый пункт u , в котором она расположена, и все другие населённые пункты, до которых можно добраться из u , используя не более k дорог (т. е. при $k = 1$, например, подстанция обслуживает свой населённый пункт и все, которые напрямую соединены с ним дорогой). Стоимость содержания такой подстанции пропорциональна её мощности.

Теперь перед руководством фирмы и, в частности, вашим другом, стоит задача придумать схему расположения подстанций в населённых пунктах области так, чтобы, во-первых, каждый населённый пункт обслуживался хотя бы одной подстанцией, а во-вторых, суммарная мощность созданных подстанций была минимальна.

Как показывает статистика, автобусы намного реже ломаются на дорогах, чем внутри населённых пунктов, где они вынуждены часто изменять скорость, останавливаться, трогаться с места, заводить двигатель и т. д., поэтому не важно, все ли дороги обслуживаются главное, чтобы обслуживались все населённые пункты.

Формат входных данных

В первой строке входного файла находится одно число N – количество населённых пунктов в области ($1 \leq N \leq 300$). Далее следуют $N - 1$ строка, описывающая дороги. Каждая строка содержит два числа – номера населённых пунктов, которые соединяет эта дорога. Населённые пункты нумеруются от 1 до N .

Формат выходных данных

В первую строку выходного файла выведите одно число – оптимальную суммарную мощность подстанций. Далее выведите N чисел, описывающих какое-нибудь оптимальное решение. i -е из этих чисел должно быть равно мощности подстанции, которую в вашем решении надо расположить в пункте i , или «0», если в населённом пункте i не должна находиться подстанция.

Пример

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

Задача I. Максимальная тройка

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

Дано дерево на n вершинах. Требуется выбрать из них три так, чтобы сумма расстояний между ними была максимальна.

Формат входных данных

Первая строка каждого теста содержит натуральное число n — количество вершин в дереве ($3 \leq n \leq 1\,000\,000$). Следующие $n - 1$ строк содержат по 2 натуральных числа v, u и описывают ребро дерева, соединяющее две вершины v и u ($1 \leq v, u \leq n$).

Формат выходных данных

Выведите единственное число — максимальную сумму расстояний.

Примеры

стандартный ввод	стандартный вывод
3 1 2 1 3	4
7 1 2 3 4 7 6 7 5 7 1 1 4	10