

Задача А. Муравьи-мутанты

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

В городе, который так бережно оберегал Человек-паук, появились муравьи. И это не просто маленькие милые насекомые. Это огромные кровожадные мутанты! Человек-паук, как обычно, не стал доверять очистку города местной полиции и решил принять удар на себя.

Известно, что муравьи-мутанты двигаются по координатной прямой. В начальный момент времени координата i -го муравья равна a_i . Каждую секунду муравьи перемещаются на одну позицию вправо. То есть, если в данный момент муравей находится в точке x , то через секунду он будет находиться в точке $x + 1$. Чтобы расправиться со злыми тварями, Человек-паук расставил ловушки на этой самой прямой. Причем i -я ловушка находится в позиции b_i . Когда муравей оказывается в точке, в которой находится ловушка, ловушка срабатывает и обездвиживает муравья. Одна ловушка может захватить не более одного муравья. Можно считать, что как только муравей попал в ловушку, эти муравей и ловушка перестают существовать.

Человеку-пауку стало интересно, в какую ловушку попал каждый муравей. Помогите Человеку-пауку, он в долгу не останется!

Формат входного файла

В первой строке входного файла даны два числа n, m ($1 \leq n \leq 100\,000, 1 \leq m \leq 100\,000$) — количество муравьев и ловушек соответственно. В следующей дано n чисел a_i ($0 \leq a_i \leq 10^9$) — координата i -го муравья. Гарантируется, что $a_i < a_{i+1}$ для всех $1 \leq i < n$. В следующей дано m чисел b_i ($0 \leq b_i \leq 10^9$) — координата i -й ловушки. Гарантируется, что $b_i < b_{i+1}$ для всех $1 \leq i < m$.

Формат выходного файла

В выходной файл выведите n строк. В i -й строке выведите номер ловушки, в которую попадет i -й муравей. Муравьи и ловушки нумеруются с единицы в том порядке, в котором они даны во входном файле. Если муравей не попадет ни в какую ловушку, в i -й строке выходного файла выведите -1 .

Пример

<code>ant.in</code>	<code>ant.out</code>
8 6	1
0 2 3 4 5 6 8 13	-1
1 3 5 6 9 12	2
	6
	3
	4
	5
	-1

Система оценивания

Первая группа тестов состоит из тестов, для которых выполняются ограничения $n \leq 100, m \leq 100$. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 50 баллов.

Вторая группа тестов состоит из тестов, для которых выполняются ограничения $n \leq 100\,000, m \leq 100\,000$. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 50 баллов.

Обратите внимание на возможность узнать результат проверки вашего решения на всех группах тестов, нажав на ссылку `Request feedback` на вкладке `Runs`.

Задача В. Разбиение на камеры

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

Вчера вечером Человек-паук неплохо поработал — поймал аж n преступников! Но тюрьмы уже переполнены, и разместить их решили на авианосце Щита. Ник Фьюри лично решил распределить преступников по камерам.

Всего на авианосце есть k камер для заключенных. Распределять преступников по камерам случайным образом Ник не хочет, поэтому он решил распределить всех преступников по камерам так, чтобы «исключающее или» числа людей во всех камерах было равно нулю. Более формально, Ник хочет в первую камеру отправить a_1 преступников, во вторую a_2 , ..., в k -ю камеру он хочет отправить a_k преступников так, чтобы выполнялись следующие условия:

- $a_1 + a_2 + \dots + a_k = n$
- $a_1 \oplus a_2 \oplus \dots \oplus a_k = 0$, где \oplus — исключающее или

Также Ник понимает, что оставлять камеры пустыми не имеет смысла, поэтому все a_i должны быть больше нуля.

Преступники уже готовят план побега, поэтому нельзя терять ни минуты. Помогите командиру авианосца как можно быстрее решить эту задачу!

Формат входного файла

В первой строке входного файла даны два числа n, k ($1 \leq n, k \leq 100\,000$) — количество преступников на корабле Щита и количество камер заключения соответственно.

Формат выходного файла

Если существует требуемое разбиение преступников по камерам, в единственной строке выходного файла выведите k положительных чисел, i -е из которых означает количество преступников, которое нужно отправить в i -ю камеру заключения.

Если нужного разбиения не существует, в единственной строке выходного файла выведите -1 . Если существует несколько разбиений, выведите любое.

Пример

<code>partition.in</code>	<code>partition.out</code>
8 2	4 4
5 3	-1

Внимание! Второй тест из условия не подходит под ограничения первой и второй групп тестов. Но Ваше решение все равно должно его проходить, чтобы получить баллы за эти группы.

Комментарий

Исключающее или — логическая операция, которая имеет следующую таблицу истинности:

- $0 \oplus 0 = 0$
- $0 \oplus 1 = 1$
- $1 \oplus 0 = 1$
- $1 \oplus 1 = 0$

«Исключающее или» чисел, состоящих из нескольких бит, считается побитово. Например, $2 \oplus 3 = 1$, $2 \oplus 5 = 7$, $5 \oplus 5 = 0$.

Более подробно про «исключающее или» можно почитать тут:
https://ru.wikipedia.org/wiki/Сложение_по_модулю_2

Система оценивания

Первая группа тестов состоит из тестов, для которых выполняется ограничение $k = 2$. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 20 баллов.

Вторая группа тестов состоит из тестов, для которых выполняется ограничение k — четное. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 30 баллов.

Третья группа тестов состоит из тестов, для которых выполняются ограничения $n, k \leq 100\,000$. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 50 баллов.

Обратите внимание на возможность узнать результат проверки вашего решения на всех группах тестов, нажав на ссылку «Request feedback» на вкладке «Runs».

Задача С. Производство паутины

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

Всем известно, что у Человека-паука есть альтер эго — Питер Паркер. Несложно догадаться, что паутину для супергероя производит как раз обычный человек, чьи карманы никогда особо не ломились от денег.

Питер хочет найти наименее затратный процесс производства. Так как формула паутины секретна, для каждой паутины существует характеризующая ее строка, состоящая из строчных латинских букв. Таким образом, чтобы приготовить паутину, нужно собрать строку из пустой, с помощью следующих операций:

1. Добавить к текущей строке любую букву. Стоимость операции равна a .
2. Приписать к текущей строке ее же. Грубо говоря, продублировать строку. Стоимость операции равна b .
3. Удалить букву из конца текущей строки. Стоимость операции равна c .

Например из строки «abc», можно получить строку «abaх», с помощью последовательности операций: 3, 2, 3, 1. Что будет суммарно стоить $a + b + 2c$.

Паркер часто меняет формулу паутины, поэтому он хочет для каждой возможной формулы знать, во сколько ему обойдется производство. Стоимость производства паутины вычисляется как суммарная стоимость всех операций, с помощью которых из пустой строки получается строка, характеризующая данную формулу паутины.

К сожалению, в городе была замечена Зловещая Шестерка, и Человеку-пауку нужно с этим разобраться. Раз уж Вы знаете его альтер эго, можно доверить Вам написание программы, которая сможет вычислить наименьшую стоимость производства паутины по ее характеризующей строке.

Формат входного файла

В первой строке входного файла содержится одно целое число n ($1 \leq n \leq 100\,000$) — длина строки, характеризующей паутину. Во второй строке входного файла дана строка длиной n , состоящая из строчных латинских букв. В третьей строке входного файла заданы три целых числа a, b, c , ($0 \leq a, b, c \leq 10^9$) — стоимость добавления одной буквы, дублирования строки и удаления одной буквы из конца соответственно.

Формат выходного файла

В единственной строке выходного файла выведите одно целое число — наименьшую суммарную стоимость операций, с помощью которых можно получить данную строку.

Пример

<code>webbuy.in</code>	<code>webbuy.out</code>
3 abc 1 1 1	3
7 abcdabc 1 2 0	6

Комментарий

В первом тесте нужно купить все буквы.

Во втором тесте нужно купить буквы «a», «b», «c», «d», потом продублировать строку и удалить «d».

Система оценивания

Первая группа тестов состоит из тестов, для которых выполняется ограничение $c = 0$ и $1 \leq n \leq 1000$. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 50 баллов.

Вторая группа тестов состоит из тестов, для которых выполняются полные ограничения. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 50 баллов.

Обратите внимание на возможность узнать результат проверки вашего решения на всех тестах, нажав на ссылку «Request feedback» на вкладке «Runs».

Задача D. Эвакуация

Имя входного файла: `evacuation.in`
Имя выходного файла: `evacuation.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Случилось ужасное — Доктор Октавиус захватил промышленную лабораторию. Лаборатория представляет собой n комнат, некоторые из которых соединены коридорами. Всего в лаборатории $n - 1$ коридоров, и известно, что из каждой комнаты можно добраться до любой другой. Иначе говоря, лаборатория представляет собой дерево с n вершинами. Для каждого коридора известно время l_i , необходимое для того, чтобы добраться из одного его конца в другой. По коридорам можно передвигаться в обоих направлениях.

В каждой комнате лаборатории находится по одному человеку. Доктор Октавиус собирается убить всех этих людей, но он дал им шанс на спасение. Лаборатория устроена так, что в листьях этого дерева находятся выходы. Листом в дереве будем называть вершину, которая имеет степень 0 или 1. Если человек добирается до выхода — он спасается из щупалец Доктора Октавиуса. В каждом коридоре злой гений установил детонатор. Для каждого детонатора известен момент времени t_i — когда он взорвется. После того как детонатор взрывается, по коридору, в котором он находился, двигаться больше невозможно, и люди, которые находились в нем этот момент, погибают. Если в момент взрыва человек добежит до конца коридора, он выживает.

Человек-паук уже не успевает прийти на помощь этим бедолагам, поэтому им нужно спастись самим. От вас же требуется выяснить — какое максимальное количество человек смогут спастись.

Формат входного файла

В первой строке входного файла дано одно число n ($1 \leq n \leq 100\,000$) — количество комнат в лаборатории. В каждой из следующих $n - 1$ строк находятся четыре числа a_i, b_i, l_i и t_i ($1 \leq a_i, b_i \leq n$; $1 \leq l_i \leq 10\,000$; $1 \leq t_i \leq 10^9$) — номера комнат, которые соединяет данный коридор, время, необходимое для того, чтобы добраться из одного его конца в другой, и момент времени, в который в этом коридоре сработает детонатор.

Гарантируется, что лаборатория представляет собой дерево.

Формат выходного файла

В единственной строке выходного файла выведите одно число — максимальное количество человек, которые смогут спастись.

Пример

<code>evacuation.in</code>	<code>evacuation.out</code>
5 3 5 2 3 1 3 3 2 4 5 2 4 2 5 1 2	5
7 1 6 4 5 3 4 3 3 5 1 4 4 7 4 2 4 2 4 4 6 3 1 3 4	6

Комментарий

В первом примере человек, который находился в комнате номер 5, бежит в комнату 4, человек,

находящийся в 3-й комнате направляется туда же, обратите внимание, что он прибывает туда в момент взрыва, но тем не менее успевает спастись.

Во втором примере человек из 4-й комнаты направляется в комнату номер 2, человек из комнаты 1 в комнату 5, а человек, находящийся в 3-й комнате не успевает спастись.

Система оценивания

Первая группа тестов состоит из тестов, для которых выполняется ограничение $n \leq 1000$. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 40 баллов.

Вторая группа тестов состоит из тестов, для которых выполняются полные ограничения. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 60 баллов.

Обратите внимание на возможность узнать результат проверки вашего решения на всех тестах, нажав на ссылку «Request feedback» на вкладке «Runs».