Задача A. Бинарная строка

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

У вас есть бинарная строка s длины n. Строка состоит из символов «0 и «1». Оказалось, что эта строка является головоломкой. Над ней можно выполнять только следующую операцию: инвертировать все биты на любом префиксе, то есть заменить все «0 на «1» и наоборот для каждого символа на некотором префиксе. В разных операциях вы можете выбирать разные префиксы.

Найдите минимальное число операций, которые нужно применить, чтобы получить строку, состоящую только из нулей.

Формат входных данных

В первой строке входных данных вводится целое число $n\,(1 \leqslant n \leqslant 10^5)$ – длина строки.

Во второй строке вводится бинарная строка s длины n.

Формат выходных данных

В единственной строке выведите ответ на задачу.

Система оценки

Правильный ответ на каждом тесте к задаче (не считая тестов из условия) оценивается в 0.4 балла.

Дополнительные ограничения на тесты (не считая тестов из условия):

1-5 тесты – $n \leqslant 10^2$.

6-15 тесты – $n \leqslant 10^3$.

16-25 тесты – без дополнительных ограничений.

Примеры

стандартный ввод	стандартный вывод
4 1010	3
5 00000	0

Задача B. Загадочный массив

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

Исследуя руины некоторого древнего города, археологи нашли огромную стену с записанным на ней массивом целых неотрицательных чисел a. Приложив немалые усилия, они смогли прочитать древние надписи и понять, что этот массив может открыть им важнейшую информацию об древней цивилизации. Одним из шагов для этого является решение некоторой задачи, связанной с этим массивом.

Над массивом можно производить следующую операцию: выбрать 2 индекса i и j $(1 \leqslant i < j \leqslant n)$, где n это длина массива a, и затем вычесть 2 из a_i и добавить 1 к a_j. После каждой операции все элементы массива должны остаться неотрицательными. Необходимо определить максимальное количество операций, которые можно применить к массиву a.

Формат входных данных

Первая строка содержит единственное целое число n $(1 \leqslant n \leqslant 10^5)$ – длину массива a.

Вторая строка содержит n чисел $a_1, a_2, \ldots, a_n$ $(0 \leqslant a_i \leqslant 10^9)$.

Формат выходных данных

В единственной строке выведите максимальное число операций, которые можно выполнить для данного массива a.

Система оценки

Правильный ответ на каждом тесте к задаче (не считая тестов из условия) оценивается в 0.2 балла.

Дополнительные ограничения на тесты (не считая тестов из условия):

1-10 тесты – $n \leqslant 10^3$ и $a_i \leqslant 2$.

11-20 тесты – $a_i \leqslant 2$.

21-30 тесты – $n \leqslant 10^3$.

31-40 тесты – $n \leqslant 10^4$.

41-50 тесты – без дополнительных ограничений.

Примеры

стандартный ввод	стандартный вывод
4 3 2 4 5	5
6 1 2 4 1 1 0	6

Задача C. Игра

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

Алиса и Боб придумали новую игру и решили сыграть в неё. Изначально есть некоторое натуральное число n. Игроки ходят по очереди. Во время своего хода игрок выбирает простое число p и некоторое натуральное число a, такие, чтобы n делилось на p^a, и затем заменяет n на $\frac{n}{p^a}$. Игрок, который не может сделать ход, проигрывает.

Первой ходит Алиса. Каким будет исход игры при оптимальной игре обоих противников?

Формат входных данных

В первой строке входных данных вводится натуральное число $t\,(1 \leqslant t \leqslant 10)$ – количество вариантов игры.

Далее в идёт t строк, в каждой из которых содержится единственное натуральное число $n\,(1 \leqslant n \leqslant 10^9)$ – число, фиксированное в начале игры для данного варианта игры.

Формат выходных данных

Для каждого варианта игры в отдельной строке выведите ответ.

Если при оптимальной игре выиграет Алиса, выведите «`Alice`», иначе выведите «`Bob`» (без кавычек).

Система оценки

Правильный ответ на каждом тесте к задаче (не считая тестов из условия) оценивается в 0.2 балла.

Дополнительные ограничения на тесты (не считая тестов из условия):

1-10 тесты – $n \leqslant 10^3$.

11-20 тесты – $n \leqslant 10^5$.

21-30 тесты – $n \leqslant 10^7$.

31-50 тесты – без дополнительных ограничений.

Примеры

стандартный ввод	стандартный вывод
2 12 360	Alice Bob
3 1 13 48	Bob Alice Alice

Задача D. Сокровище

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

Спустя годы поисков, разочарований и приключений кладоискатели нашли легендарную пещеру, где должен был быть зарыт огромный клад. Однако здесь они не нашли сокровище, а лишь k ключей от сундуков и загадочное послание. Оказалось, что клад зарыт в n различных городах, которые соединяет одна железная дорога. Причём в городе i спрятано a_i монет.

В данный момент кладоискатели находятся в городе 1. Для того чтобы перемещаться между городами, они будут пользоваться железной дорогой. Однако им придётся платить c_i монет, чтобы из города i добраться до города $i+1$ – других путей между городами нет.

Кладоискатели в сумме могут открыть не более k сундуков, но если они приезжают в город i, они не обязательно будут тратить свой ключ на клад здесь. После того как они прекратят поиски сокровища, они смогут поехать обратно бесплатно (в качестве бонуса от железнодорожной компании).

Необходимо определить, какую максимальную прибыль они смогут получить.

Формат входных данных

В первой строке содержатся два целых числа n, k $(1 \leqslant k \leqslant n \leqslant 10^5)$.

Во второй строке содержатся n целых чисел $a_1, a_2, \ldots, a_n$ $(0 \leqslant a_i \leqslant 10^9)$ – количество монет в каждом городе.

Во третьей строке содержатся $n-1$ целых числа $_1, _2, \ldots, _{n-1}$ $(0 \leqslant c_i \leqslant 10^9)$ – стоимости перемещения из города i в город $i+1$.

Формат выходных данных

Выведите максимально возможную прибыль.

Система оценки

Правильный ответ на каждом тесте к задаче (не считая тестов из условия) оценивается в 0.2 балла.

Дополнительные ограничения на тесты (не считая тестов из условия):

1-20 тесты – $n \leqslant 10^3$.

21-30 тесты – $n \leqslant 10^4$.

31-50 тесты – без дополнительных ограничений.

Примеры

стандартный ввод	стандартный вывод
5 2 10 3 2 20 45 5 5 5 50	15
7 3 0 0 5 8 13 17 20 9 5 7 10 1 8	10

Задача E. Лабиринт времени

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	6 секунд
Ограничение по памяти:	256 мегабайт

Некоторая лаборатория состоит из n комнат, соединённых m двусторонними коридорами, для прохождения которых требуется некоторое количество времени. В лаборатории проводились загадочные эксперименты со временем, в результате чего оказалось, что если пройти по некоторому маршруту и посетить при этом какие-то пары коридоров, потраченное время увеличится на некоторую величину, хотя на первый взгляд этого не должно происходить. Главный учёный узнал об этом, и поэтому собирается отправиться в основную комнату лаборатории n, чтобы исправить аномалии. Изначально он находится в комнате 1.

Помогите ему определить, какое наименьшее время ему для этого понадобится. Если добраться до комнаты n из комнаты 1 невозможно, выведите -1.

Формат входных данных

В первой строке содержатся 3 целых числа n, m и k ($1 \leqslant n \leqslant 10^5$, $1 \leqslant m \leqslant 10^5$, $0 \leqslant k \leqslant 5$) – количество комнат в лаборатории, количество коридоров и количество пар особых коридоров.

В следующих m строках содержатся по 3 числа u, v и t, означающие, что между комнатами u и v есть двусторонний проход, на прохождение которого требуется t времени ($1 \leqslant u, v \leqslant n$, $1 \leqslant t \leqslant 10^9$).

Далее в k строках содержатся по 3 числа i, j и t ($1 \leqslant i, j \leqslant m$, $i \neq j$, $1 \leqslant t \leqslant 10^9$), означающие, что если на пути от 1 до n учёный посетит и коридор номер i, и коридор номер j (в любом порядке), то время его пути увеличится на t.

Гарантируется, что все номера $2 \cdot k$ коридоров, увеличивающих время, различны.

Формат выходных данных

В единственной строке выведите ответ на задачу.

Система оценки

Правильный ответ на каждом тесте к задаче (не считая тестов из условия) оценивается в 0.2 балла.

Дополнительные ограничения на тесты (не считая тестов из условия):

1-10 тесты – $k = 0$.

11-30 тесты – $k \leqslant 2$.

31-50 тесты – без дополнительных ограничений.

Примеры

стандартный ввод	стандартный вывод
5 6 2 1 2 2 2 3 6 3 5 3 2 5 7 1 4 5 4 5 1 5 6 10 1 3 10	9
5 6 3 1 2 2 2 3 6 3 5 3 2 5 7 1 4 5 4 5 1 5 6 14 1 4 100 2 3 10	20

Задача A. Бинарная строка

Заметим, что, если $s_i = s_{i+1}$ ($s_i \neq s_{i+1}$), то это равенство (неравенство) может измениться только при применении операции к префиксу, у которого последний символ это s_i. Значит, необходимо применить операцию для каждого i, для которого $s_i \neq s_{i+1}$. После этого строка будет состоять из одинаковых символов. Если на конце «1», то необходимо применить ещё 1 операцию, чтобы инвертировать всю строку.

Задача B. Загадочный массив

При применении любой операции мы получаем возможность добавить единицу к любому элементу с большим индексом – после выполнения каждой операции будем добавлять 1 единицу, которую мы можем прибавить далее в некоторый счётчик. Рассмотрим элементы с начала до конца и будем хранить число единиц, которое мы можем добавить к любому элементу далее. Если очередной элемент больше 1, то будем вычитать из него 2, пока можем (и для последующих операций запомним, сколько раз мы сможем добавить единицу за счёт этого). Это является оптимальным, так как далее мы не сможем использовать этот элемент никак. Если он после этого равен 1 и мы можем добавить к нему 1, добавим и выполним ещё операцию – это также выгодно, так как мы получаем то же 1 прибавление назад и выполняем дополнительную операцию. Теперь рассмотрим случай, когда элемент равен 0, но мы можем добавить ещё 2 единицы, чтобы выполнить операцию. Оказывается, что в этом случае добавление 2 единиц является оптимальным.

Рассмотрим все единицы, которые мы будем куда-либо добавлять. Они либо не принесут никаких операций, либо принесут 1 операцию, комбинируясь с единичкой, первоначально находившейся в некотором месте, либо принесут 1 операцию, комбинируясь с другой прибавленной единичкой. Однако второй из этих случаев возможен лишь тогда, когда мы, рассматривая новую позицию в нашем алгоритме, рассматриваем нечётное число и при этом у нас есть хотя бы 1 единица, которую можно применить. Все остальные добавления единиц должны комбинироваться между собой – при этом мы тратим 2 операции прибавления и получаем одну, а значит, вне зависимости от того, когда мы будем тратить единицы таким образом, мы не будем никак влиять на возможность потратить единицы способом 2. Тогда выгоднее тратить их как можно раньше, так как полученные прибавления при этом можно будет применить на больший участок массива.

Таким образом, алгоритм, описанный выше, является корректным.

Задача C. Игра

Для решения введём функцию от n, которая равна 1, если для этого n делающий ход игрок выигрывает, иначе 0. Тогда если мы для некоторого n можем перейти в проигрышное состояние, значение функции 1, иначе – 0 (так как для любого хода в этом случае у соперника есть выигрышная стратегия).

Количество делителей n не превышает $\mathcal{O}(\sqrt{n})$, а число переходов не превышает числа различных простых делителей числа (можно грубо оценить как $\mathcal{O}(\log n)$), умноженного на максимальную степень простого числа (также не превышает $\mathcal{O}(\log n)$). Для каждого n переходы рассмотрим только 1 раз, а затем запомним значение функции для данного n. Тогда получим решение с асимптотикой $\mathcal{O}(\sqrt{n}\log^2 n)$, что является достаточным.

Задача D. Сокровище

Пусть кладоискатели остановятся в городе i, тогда наибольшая прибыль, которую они получат, это сумма k максимумов префикса массива a до i включительно минус сумма массива c до i не включительно. Ответом будет наибольший ответ среди этих случаев. Тогда для решения достаточно поддерживать k максимумов на префиксе, что можно сделать, используя структуры, поддерживающие поиск минимума, добавление и удаление (`std::multiset` в C++). Такое решение будет работать за $\mathcal{O}(n \log n)$.

Задача E. Лабиринт времени

Если бы в задаче не было k дополнительных добавлений времени, то задачу можно было бы решить обычным алгоритмом Дейкстры. Для того, чтобы учитывать дополнительные k условий, будем использовать пары $(u, mask)$ для алгоритма Дейкстры. Такая пара будет означать минимальное расстояние до вершины u, если при этом мы посетили хотя бы 1 ребро из условия i, $0 \leqslant i \leqslant k-1$ для всех i таких, что бит под номером i в маске равен «1». Тогда при переходе из такой пары нужно, как и в графе, перебрать исходящие из u рёбра и соответствующие конечные вершины v. Если эти рёбра не участвуют в условиях, то переходим в $(v, mask)$. Если ребро участвует в условии, то нужно либо добавить «1» в соответствующий бит маски, чтобы далее при возможном посещении второго ребра учесть дополнительное условие, либо, если там была единица, добавить к потраченному времени время из соответствующего условия.

В таком графе мы будем рассматривать $\mathcal{O}(m2^k)$ переходов, а значит асимптотика решения будет $\mathcal{O}(m2^k(\log m + k))$.