

Задача А. Выбор вершин взвешенного дерева

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

Дан граф, являющийся деревом. В вершинах графа написаны целые числа. Множество вершин графа называется *допустимым*, если никакие две вершины этого множества не соединены ребром.

Рассмотрим все допустимые множества вершин графа. Для каждого такого множества вычислим сумму чисел, написанных в его вершинах. Какова максимальная из этих сумм?

Формат входных данных

Граф в этой задаче задан в виде *корневого дерева*. В графе выделена вершина — *корень дерева*. Для каждой вершины i , не являющейся корнем, задан номер вершины-предка p_i в корневом дереве. Дерево, заданное таким образом, состоит из рёбер $i - p_i$ для всех вершин i , кроме корня.

В первой строке входного файла записано целое число n — количество вершин в графе ($1 \leq n \leq 100$). В следующих n строках задан граф. В i -й из этих строк записаны через пробел два целых числа p_i и q_i ; здесь p_i — номер вершины-предка i -ой вершины, а q_i — число, записанное в этой вершине. Для корня дерева $p_i = 0$; для всех остальных вершин $1 \leq p_i \leq n$. Числа q_i не превосходят по модулю 10 000.

Гарантируется, что заданный во входном файле граф является деревом.

Формат выходных данных

В первой строке выходного файла выведите одно число — максимальную сумму чисел в допустимом множестве.

Примеры

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

Задача В. 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 С. Дубы

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

На аллее перед зданием Министерства Обороны в ряд высажены n дубов. В связи с грядущим приездом главнокомандующего, было принято решение срубить несколько деревьев для придания аллее более милитаристического вида.

Внутренние распоряжки министерства позволяют срубить дуб только в двух случаях:

- если и ближайший дуб слева, и ближайший дуб справа строго ниже, чем данный дуб;
- если и ближайший дуб слева, и ближайший дуб справа строго выше, чем данный дуб.

В частности, согласно этому правилу, нельзя срубить крайний левый и крайний правый дубы.

Министр хочет выработать такой план вырубki, чтобы в итоге осталось несколько дубов, высоты которых образуют неубывающую последовательность, то есть чтобы каждый дуб был не ниже, чем все дубы, стоящие слева от него. При этом, как человек любящий флору, министр хочет, чтобы было срублено минимальное возможное количество деревьев.

Помогите сотрудникам министерства составить оптимальный план вырубki аллеи или выяснить, что срубить дубы соответствующим образом невозможно.

Формат входных данных

Первая строка входного файла содержит целое число n — количество дубов, растущих на аллее ($2 \leq n \leq 200$). Вторая строка содержит n чисел — высоты дубов, приведенные слева направо. Высоты дубов — положительные целые числа, не превышающие 1 000.

Формат выходных данных

Если оставить последовательность дубов с неубывающими высотами невозможно, выходной файл должен содержать только одно число -1 .

В случае, если искомый план существует, в первую строку выходного файла выведите целое число m — минимальное количество дубов, которые необходимо срубить. В следующие m строк выведите оптимальный план вырубki деревьев — номера дубов в том порядке, в котором их следует срубить, по одному номеру на строке.

Дубы нумеруются слева направо натуральными числами от 1 до n .

Если планов с наименьшим числом срубаемых дубов несколько, выведите любой из них.

Пример

стандартный ввод	стандартный вывод
5	2
3 2 4 8 5	4
	2

Задача D. Удаление скобок

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 8 мегабайт

Обратите внимание на необычное ограничение по памяти.

Дана строка, составленная из круглых, квадратных и фигурных скобок. Определите, какое наименьшее количество символов необходимо удалить из этой строки, чтобы оставшиеся символы образовывали правильную скобочную последовательность.

Формат входных данных

Входные данные состоят из строки из круглых, квадратных и фигурных скобок. Длина строки не превосходит 700 символов.

Формат выходных данных

Выведите строку максимальной длины, являющуюся правильной скобочной последовательностью, которую можно получить из исходной строки удалением некоторых символов.

Если возможных ответов несколько, выведите любой из них.

Примеры

стандартный ввод	стандартный вывод
([])	[]
{([(){}])}	[({})]

Задача E. Оптимальное бинарное дерево поиска

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

Рассмотрим множество $S = e_1, e_2, \dots, e_n$, состоящее из n различных элементов таких, что $e_1 < e_2 < \dots < e_n$. Рассмотрим бинарное дерево поиска, состоящее из элементов S . Чем чаще производится запрос к элементу, тем ближе он должен располагаться к корню.

Стоимостью $cost$ доступа к элементу e_i из S в дереве будем называть значение $cost(e_i)$, равное числу ребер на пути, который соединяет корень с вершиной, содержащей элемент. Имея частоты запросов к элементам из S , $(f(e_1), f(e_2), \dots, f(e_n))$, определим общую стоимость дерева следующим образом:

$$f(e_1) \cdot cost(e_1) + f(e_2) \cdot cost(e_2) + \dots + f(e_n) \cdot cost(e_n)$$

Дерево, имеющее наименьшую стоимость, считается наилучшим для поиска элементов из S . Именно поэтому оно называется Оптимальным Бинарным Деревом Поиска.

Формат входных данных

Состоит из нескольких тестов, каждый из которых расположен в отдельной строке.

Первое число в строке n ($1 \leq n \leq 250$) указывает на размер множества S . Следующие n неотрицательных целых чисел описывают частоты запросов элементов из S : $f(e_1), f(e_2), \dots, f(e_n)$. Известно, что $0 \leq f(e_i) \leq 100$.

Сумма n по всем тестам не больше 5000.

Формат выходных данных

Для каждого теста в отдельной строке вывести стоимость Оптимального Бинарного Деревя Поиска.

Пример

стандартный ввод	стандартный вывод
1 5	0
3 10 10 10	20
3 5 10 20	20

Задача F. Распил брусьев

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

Вам нужно распилить деревянный брус на несколько кусков в заданных местах. Распилочная компания берет k рублей за распил одного бруска длиной k метров на две части.

Понятно, что различные способы распила приводят к различной суммарной стоимости заказа. Например, рассмотрим брус длиной 10 метров, который нужно распилить на расстоянии 2, 4 и 7 м, считая от одного конца. Это можно сделать несколькими способами. Можно распилить сначала на отметке 2 м, потом 4 и, наконец, 7 м. Это приведет к стоимости $10+8+6=24$, потому что сначала длина бруса, который пилили, была 10 м, затем она стала 8 м, и, наконец, 6 м. А можно распилить иначе: сначала на отметке 4 м, затем 2, затем 7 м. Это приведет к стоимости $10+4+6=20$, что лучше.

Определите минимальную стоимость распила бруса на заданные части.

Формат входных данных

Первая строка входных данных содержит целое число L ($2 \leq L \leq 10^6$) - длину бруса и целое число N ($1 \leq N \leq 100$) - количество распилов. Во второй строке записано N целых чисел C_i ($0 < C_i < L$) в строго возрастающем порядке - места, в которых нужно сделать распилы.

Формат выходных данных

Выведите одно натуральное число - минимальную стоимость распила.

Пример

стандартный ввод	стандартный вывод
10 3 2 4 7	20

Задача G. Почти палиндромы

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

Слово называется *палиндромом*, если его первая буква совпадает с последней, вторая – с предпоследней и т.д. Например, слова «abba», «madam», «x» являются палиндромами.

Для заданного числа K слово называется *почти палиндромом*, если в нём можно изменить не более K букв так, чтобы получился палиндром. Например, при $K = 2$ слова «reactor», «kolobok», «madam» являются почти палиндромами, так как могут быть преобразованы в «reacaer», «kololok», «madam» заменой двух, одной и нуля букв соответственно.

Подсловом данного слова являются все непустые слова, получающиеся путем вычеркивания из данного нескольких (возможно, нуля) первых букв и нескольких последних. Например, подсловами слова «cat» являются слова «с», «а», «t», «са», «at» и само слово «cat» (а «ct» подсловом слова «cat» не является).

Требуется для данного числа K определить, сколько подслов данного слова S являются почти палиндромами.

Формат входных данных

В первой строке вводятся два натуральных числа: N ($1 \leq N \leq 5 \cdot 10^3$) – длина слова и K ($0 \leq K \leq N$).

Во второй строке содержится слово S , состоящее из N строчных латинских букв.

Формат выходных данных

Требуется вывести одно число – количество подслов слова S , являющихся почти палиндромами (для данного K).

Примеры

стандартный ввод	стандартный вывод
5 1 abcde	12
3 3 aaa	6

Задача Н. Этично ли удалять рёбра у связанного графа?

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

Граф Безциклов решил проверить своё здоровье. Он хочет проверить, что все его рёбра достаточно крепко держатся в нём. Для этого он хочет посчитать *устойчивость* некоторых из них. *Устойчивостью* ребра называется количество простых путей, проходящих через это ребро.

Формат входных данных

Все числа в файле целые.

$0 \leq N \leq 10^5$, $0 \leq M \leq 10^5$ — количество вершин и рёбер.

Затем M пар чисел $1 \leq v_i, u_i \leq N$ — i -ое ребро соединяет вершины v_i и u_i .

$0 \leq Q \leq 10^5$ — количество запросов.

Затем Q чисел $1 \leq e_i \leq M$.

Граф неориентирован. Гарантируется, что Граф ацикличесен.

Формат выходных данных

Для i -ого запроса вывести устойчивость e_i -ого ребра.

Пример

стандартный ввод	стандартный вывод
2 1 1 2 1 1	1

Задача I. ЧСВ - чувство собственной важности.

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

Граф Безциклов считает себя очень важной и значимой личностью. Чтобы это доказать, он решил вычислить свою *важность* и *значимость*. *Важность* Графа — количество простых путей в нём, его *значимость* — сумма длин этих путей.

Формат входных данных

Все числа в файле целые.

$0 \leq N \leq 10^5$, $0 \leq M \leq 10^5$ — количество вершин и рёбер.

Затем M пар чисел $1 \leq v_i, u_i \leq N$ — i -ое ребро соединяет вершины v_i и u_i .

Граф неориентирован. Гарантируется, что Граф ацикличен.

Формат выходных данных

Вывести важность и значимость Графа.

Пример

стандартный ввод	стандартный вывод
2 1	1
1 2	1