

Задача А. Дерево деревьев

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

Однажды ночью, когда Артём и Андрей торопились в баню, они вспомнили одну историю. Шёл как-то Сергей Андреевич по просторам Берендеевых полей и увидел дерево, высокое-превысокое. Он посмотрел на него и задумался, а сколько других деревьев может быть на нём? Ребята посмеялись над этой историей, ведь на дереве не может быть других деревьев. Но им в голову пришла гениальная идея - сделать собственное уникальное дерево деревьев. Вы не поверите, они его сделали!!! Но теперь перед ними встал вопрос, а сколько же деревьев произрастает на их поделке? Итак, дано дерево. Нужно посчитать количество множеств вершин этого дерева, которые сами образуют дерево

Формат входных данных

В первой строке входного файла находится число n ($1 \leq n \leq 2 \cdot 10^5$) - количество вершин. В следующих $n - 1$ строках описываются веточки в их поделке. Каждая веточка задаётся парой чисел a, b ($1 \leq a \leq n, 1 \leq b \leq n, a \neq b$). Гарантируется, что у ребят получилось именно дерево

Формат выходных данных

Выходной файл должен содержать ответ на задачу - число различных непустых множеств вершин исходного дерева, которые сами образуют дерево. Так как ответ может быть достаточно большим, нужно вывести его по модулю $10^9 + 7$

Примеры

стандартный ввод	стандартный вывод
4 1 2 1 3 3 4	10
7 1 2 1 3 2 4 2 5 3 6 3 7	37

Задача В. Кратчайший путь

Имя входного файла: dag-shortpath.in
Имя выходного файла: dag-shortpath.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан ориентированный взвешенный ациклический граф. Требуется найти в нем кратчайший путь из вершины s в вершину t .

Формат входных данных

Первая строка входного файла содержит четыре целых числа n , m , s и t — количество вершин, дуг графа, начальная и конечная вершина соответственно. Следующие m строк содержат описания дуг по одной на строке. Ребро номер i описывается тремя целыми числами b_i , e_i и w_i — началом, концом и длиной дуги соответственно ($1 \leq b_i, e_i \leq n$, $|w_i| \leq 1000$).

Входной граф не содержит циклов и петель.

$1 \leq n \leq 100\,000$, $0 \leq m \leq 200\,000$, $1 \leq s, t \leq n$.

Формат выходных данных

Первая строка выходного файла должна содержать одно целое число — длину кратчайшего пути из s в t . Если пути из s в t не существует, выведите `Unreachable`.

Примеры

dag-shortpath.in	dag-shortpath.out
2 1 1 2 1 2 -10	-10
2 1 2 1 1 2 -10	Unreachable

Задача С. Красота фейерверка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

В лаборатории теоретической пиротехники изучают новые технологии организации фейерверков. Фейерверк представляется как корневое дерево, а поскольку в мощном фейерверке его элементы также взрываются, порождая новые фейерверки, то ученые вводят операцию возведения корневого дерева в степень.

Корневое дерево содержит одну или несколько вершин. Одна из вершин выделена и называется корнем дерева, для каждой из остальных вершин ровно одна другая вершина является родителем. При этом от любой вершины можно Добраться До корня, последовательно переходя от вершины к ее родителю. Вершина, которая не является родителем никакой другой вершины, называется листом. Если вершина x является родителем вершины y , то вершина y является ребенком вершины x . Будем говорить, что вершина и ее родитель соединены ребром.

На рис. 1 показан пример корневого дерева с корнем в вершине 1. Родителем вершин 2 и 3 является вершина 1, родителем вершины 4 является вершина 2. Вершины 2 и 3 – дети вершины 1, а вершина 4 – ребенок вершины 2. Листьями являются вершины 3 и 4.

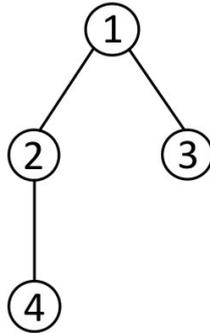


Рис. 1: Пример корневого дерева с корнем в вершине 1, листьями 3 и 4

Фейерверк задается своим базовым деревом T и мощностью m . Фейерверк представляется деревом, которое получается в результате возведения дерева T в степень m . Операция возведения дерева в степень устроена следующим образом. Если $m = 1$, то результат T^1 – само дерево T . Для $m > 1$ рассмотрим дерево T^{m-1} . Выполним следующую операцию: для каждого листа x дерева T^{m-1} создадим копию дерева T и назначим лист x родителем корня соответствующей копии. Получившееся дерево будет деревом T^m .

На рис. 2 показано дерево, представленное на рис. 1, в степенях 1, 2 и 3.

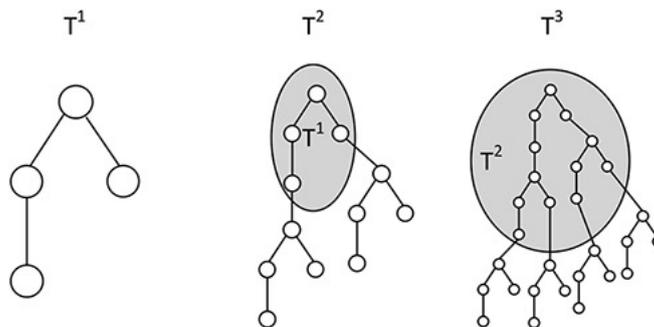


Рис. 2: Пример возведения дерева в степени 1, 2 и 3

Путем в дереве называется последовательность вершин, в которой две соседние вершины соединены ребром. Все вершины в пути должны быть различны.

Для того, чтобы оценить красоту фейерверка, необходимо определить, какое максимальное количество вершин может содержать путь в дереве, которым представляется фейерверк. На рис. 3 приведен путь в дереве T^2 , содержащий максимальное количество вершин. Таким образом, красота фейерверка с базовым деревом T и мощностью 2 равна 10.

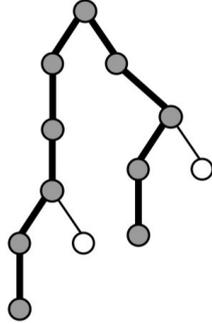


Рис. 3: Путь в дереве T^2 , содержащий максимальное количество вершин

Требуется написать программу, которая по описанию дерева T и натуральному числу m определяет красоту фейерверка с базовым деревом T и мощностью m .

Формат входных данных

В первой строке задано число n ($3 \leq n \leq 200\,000$) и число m ($1 \leq m \leq 200\,000$) — размер дерева и степень, в которую его требуется возвести соответственно.

Дальше следует строка, содержащая $n - 1$ число: p_2, \dots, p_n — предки соответствующих вершин дерева, $1 \leq p_i < i$

Корнем дерева является вершина с номером 1, гарантируется, что она не является листом.

Формат выходных данных

Выведите одно число — число вершин в диаметре дерева.

Примеры

стандартный ввод	стандартный вывод
4 2 1 1 2	10

Задача D. Пастбища

Имя входного файла: `pasture.in`
Имя выходного файла: `pasture.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Фермер Джон решил снабдить каждую из своих коров сотовым телефоном. Для этого ему требуется установить сотовые станции на его N пастбищах, последовательно пронумерованных от 1 до N .

Ровно $N - 1$ пара пастбищ являются соседними, и для любых двух пастбищ A и B имеется последовательность соседних пастбищ таких, что A — первое пастбище этой последовательности, а B — последнее. Сотовые станции размещаются только на пастбищах и имеют достаточный радиус действия, чтобы обеспечить связью это пастбище и все соседние.

Помогите фермеру Джону определить, какое минимальное количество станций он должен установить, чтобы обеспечить связью все пастбища.

Формат входных данных

На первой строке входного файла находится одно целое число N ($1 \leq N \leq 100\,000$). Далее следуют $N - 1$ строк, каждая из которых содержит пару целых чисел — очередную пару соседних пастбищ A и B ($1 \leq A \leq N$; $1 \leq B \leq N$; $A \neq B$).

Формат выходных данных

Выведите в выходной файл одно число — минимальное достаточное количество станций.

Примеры

<code>pasture.in</code>	<code>pasture.out</code>
5 1 3 5 2 4 3 3 5	2

Задача Е. Контест-палиндром

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Как вы знаете, Хет и Юра очень любят готовить контесты и готовы заниматься этим целыми часами. Вот и сейчас Хет перебирает задачи, которые он хочет добавить в предстоящий контест. Он считает, что контест будет **успешным** только в том случае, если первые буквы названий задач, которые в него входят, образуют палиндром (контест, который не содержит ни одной задачи, по логике Хета, тоже успешный!).

Система подготовки контестов «Моногон» содержит в себе N задач, которые пронумерованы последовательно от 1 до N . Все их названия начинаются с заглавной латинской буквы. Хет хочет выбрать некоторые из них. При этом, «Моногон» требует, чтобы задачи в контесты были отсортированы по порядковому номеру. Т. е. чтобы номера задач в контесте образовывали возрастающую последовательность.

Хет выписал первые буквы названий задач с номера 1 по N и принёс вам. Помогите Хету определить, сколько всего **успешных** контестов он сможет подготовить.

Так как количество контестов может быть очень большим, ответ требуется вывести по модулю $10^9 + 7$.

Формат входных данных

Единственная строка представляет из себя последовательность длины N ($1 \leq N \leq 5000$) из латинских заглавных букв.

Формат выходных данных

Выведите кол-во успешных контестов по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
АВАСАВА	42
ІТМО	5

Замечание

В первом примере название задачи №1 начинается с буквы «А», №2 с буквы «В», №3 с буквы «А» и т. д.

Задача F. Удаление скобок - 2

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

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

Во входном файле записана строка из круглых, квадратных и фигурных скобок. Длина строки не превосходит 100 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

<code>stdin</code>	<code>stdout</code>
<code>([])</code>	<code>[]</code>
<code>{([[]{}])}</code>	<code>([]{})</code>
<code>]{}[</code>	

Задача G. Распил брусьев

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
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 расстояниях 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.

Примеры

<code>stdin</code>	<code>stdout</code>
10 3 2 4 7	20
100 3 15 50 75	200

Задача Н. Игра

Имя входного файла: `game.in`
Имя выходного файла: `game.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

На листке записано в одну строку $N(2 \leq N \leq 100)$ целых положительных чисел. Каждое число не превышает 200. Играют двое. За каждый ход можно зачеркивать крайнее число либо слева, либо справа. Зачеркнутое число добавляется к очкам игрока. N – четное. Игру начинает первый игрок. Необходимо вывести максимально возможную сумму очков для первого игрока при условии, что противник играет наилучшим образом.

Формат входных данных

В первой строке входного файла содержится одно целое число $N(2 \leq N \leq 100)$. В следующих N строках записан исходный ряд чисел, по одному числу в строке.

Формат выходных данных

Выходной файл должен содержать единственное число – максимально возможную сумму очков для первого игрока при наилучшей игре второго игрока.

Примеры

<code>game.in</code>	<code>game.out</code>
4	16
4	
7	
2	
9	

Задача I. Покраска забора

Имя входного файла: `paint.in`
Имя выходного файла: `paint.out`
Ограничение по времени: 0.7 секунда
Ограничение по памяти: 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

<code>paint.in</code>	<code>paint.out</code>
5 ABBCA	3 1 5 A 4 4 C 2 3 B
2 AA	1 1 2 A

Задача J. 17 стульев

Имя входного файла: `trader.in`
Имя выходного файла: `trader.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Остап Бендер снова пытается получить причитающиеся драгоценности, но на этот раз они были заперты в шкатулке, для открытия которой необходимо иметь N ключей. По закономерной случайности каждый из ключей был спрятан в одном из N стульев, распроданных на недавнем аукционе. После аукциона эти стулья были развезены в N городов.

И вот теперь Остап решил на новую безумную затею: заехать в каждый из городов и, провернув в каждом из них аферу, выкрасть необходимые ключи. Чтобы избежать конфликтов с недоброжелателями, Остап не хочет больше одного раза появляться в каком-либо городе. Также у Остапа есть список цен за проезд между каждой парой городов. Изначально Остап находится в городе под номером 1 и после посещения всех городов может незаметно скрыться из этой страны.

Помогите Остапу найти порядок посещения городов, при котором ему потребуется потратить как можно меньше средств на странствия, и тогда, возможно, он поделится с Вами добытыми бриллиантами.

Формат входных данных

Первая строка содержит единственное число N — количество городов ($1 \leq N \leq 17$).

Следующие N строк содержат по N целых неотрицательных чисел. j -тое число в i -той строке означает стоимость проезда из города i в город j ($0 \leq a_{ij} \leq 100$). Если $a_{ij} > 0$, то проезд стоит a_{ij} рублей, иначе — это означает, что из города i в j невозможно проехать напрямую.

Формат выходных данных

В первой строке выведите минимальную сумму денег, необходимую для посещения всех городов Остапом. В следующей строке выведите N чисел — порядок посещения городов, при котором эта сумма достигается. Если затею Остапа невозможно вывести, то в единственной строке выходного файла выведите число -1.

Примеры

<code>trader.in</code>	<code>trader.out</code>
3 0 3 2 3 0 6 2 6 0	8 1 3 2
5 0 6 4 0 0 6 0 7 0 7 4 7 0 0 0 0 0 0 0 2 0 7 0 2 0	20 1 3 2 5 4

Задача К. Раскраска графа

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

Дан граф из n вершин, раскрасьте его в минимально возможное число цветов так, чтобы никакие две вершины, соединенные ребром, не были одного цвета.

Формат входных данных

В первой строке содержится число t — количество тестовых примеров ($1 \leq t \leq 5$).

Далее содержится t тестовых случаев, заданных в следующем формате:

В первой строке записаны числа n и m — количество вершин и ребер соответственно ($1 \leq n \leq 17$, $0 \leq m \leq \frac{n \cdot (n-1)}{2}$).

Затем идет m строк, в которых содержится по два числа v_i u_i , что означает, что вершины v_i и u_i соединены ребром ($1 \leq v_i, u_i \leq n, v_i \neq u_i$).

Гарантируется, что все ребра в каждом тестовом случае различны.

Формат выходных данных

Для каждого тестового случая в первой строке выведите минимальное число цветов k .

Во второй строке выведите n чисел a_i — цвета вершин ($1 \leq a_i \leq k$).

Примеры

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

Задача L. Разносчик пиццы

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

Вы подрабатываете разноской пиццы. У вас есть рюкзак размера S и огромный заказ на n пицц, i -я из которых имеет размер a_i . Разумеется, доставить пиццу требуется как можно скорее. К сожалению, у вас нет ни машины, ни друзей, которые могли бы помочь, так что единственный способ перевозки — распределить все пиццы в стопки размера не более S каждая и доставлять стопки по очереди. Вам надо распределить все пиццы из заказа в минимально возможное количество стопок.

Формат входных данных

Входной файл состоит из t тестов ($1 \leq t \leq 10$). Первая строка файла содержит число t , далее следуют описания тестов. Каждый тест описывается двумя строчками: на первой располагаются целые числа n ($1 \leq n \leq 20$) и S ($1 \leq S \leq 10^9$), на второй располагаются целые числа a_1, a_2, \dots, a_n ($1 \leq a_i \leq S$).

Формат выходных данных

Для каждого теста выведите на отдельной строке минимальное число стопок m , а на следующих m строчках — описание стопок. i -я из последующих строк должна содержать количество пицц в i -й стопке k_i и список из k_i номеров пицц. Каждая пицца должна встречаться ровно в одной стопке. Если есть несколько оптимальных решений, выведите любое из них.

Примеры

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

Задача М. Сокровища

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

Дочь короля Флатландии собирается выйти за прекрасного принца. Принц хочет подарить принцессе сокровища, но он не уверен какие именно бриллианты из своей коллекции выбрать.

В коллекции принца n бриллиантов, каждый характеризуется весом w_i и стоимостью v_i . Принц хочет подарить наиболее дорогие бриллианты, однако король умен и не примет бриллиантов суммарного веса больше R . С другой стороны, принц будет считать себя жадным всю оставшуюся жизнь, если подарит бриллиантов суммарным весом меньше L .

Помогите принцу выбрать набор бриллиантов наибольшей суммарной стоимости, чтобы суммарный вес был в отрезке $[L, R]$.

Формат входных данных

Первая строка содержит число n ($1 \leq n \leq 32$), L и R ($0 \leq L \leq R \leq 10^{18}$). Следующие n строк описывают бриллианты и содержит по два числа — вес и стоимость соответствующего бриллианта ($1 \leq w_i, v_i \leq 10^{15}$).

Формат выходных данных

Первая строка вывода должна содержать k — количество бриллиантов, которые нужно подарить принцессе. Вторая строка должна содержать номера даримых бриллиантов.

Бриллианты нумеруются от 1 до n в порядке появления во входных данных.

Если составить подарок принцессе невозможно, то выведите 0 в первой строке вывода.

Примеры

dowry.in	dowry.out
3 6 8	1
3 10	2
7 3	
8 2	