

Содержание

Must have	2
Задача 24А. Сортировка строк [0.1, 256]	2
Задача 24В. Пути нужной длины (easy) [0.2, 256]	3
Обязательные задачи	4
Задача 24С. Центроиды дерева [1.2, 256]	4
Задача 24D. БДБД [1, 256]	5
Задача 24Е. Сбалансированные строки [0.6, 256]	6
Для искателей острых ощущений	7
Задача 24F. Строки в дереве [1.2, 256]	7
Для мастеров AI	8
Задача 24G. Приятная прогулка [1, 256]	8
Задача 24H. Пути нужной длины (hard) [0.9, 256]	10

У вас не получается читать/выводить данные?

Воспользуйтесь примерами (c++) (python).

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

Обратите внимание на GNU C++ компиляторы с суффиксом inc.

Подни можно пользоваться **дополнительной библиотекой** (optimization.h).

То есть, использовать быстрый ввод-вывод: **пример про числа и строки**.

И быструю аллокацию памяти (ускоряет vector-set-map-весь-STL): **пример**.

Для тех, кто хочет разобраться, как всё это работает.

Короткая версия быстрого ввода-вывода (**тык**) и короткая версия аллокатора (**тык**).

Must have

Задача 24А. Сортировка строк [0.1, 256]

Рассмотрим задачу: даны n строк, склеить их в таком порядке, чтобы результат был лексикографически минимален.

Ваша задача – построить тест к следующим неверным жадностям:

1. `less(s,t): s < t.`
2. `less(s,t): s+s < t+t.`

Кстати, верная жадность выглядит так:

`less(s,t): s+t < t+s.`

Формат входных данных

Тип жадности – 1 или 2.

Формат выходных данных

Тест, на котором жадность даст неверный ответ.

В первой строке n ($2 \leq n \leq 10$). Далее n строк длины от 1 до 100 из букв «a» и «b» латинского алфавита.

Примеры

stdin	stdout
1	3 aaa a aa

Замечание

Пример лишь демонстрирует формат вывода, в примере приведён неверный ответ.

Подсказка по решению

В C++ мы можем просто передать функции `sort` оператор меньше, приведённый в условии. В python мы можем использовать только `a.sort(key=?)`, чтобы получить правильный `key` используют обёртку `a.sort(key=functools.cmp_to_key(f))`, где `f` – функция от двух аргументов, возвращающая $-1, 0, 1$ в зависимости от результата сравнения. А ещё, если не получается, можно отсортировать пузырьком.

Задача 24В. Пути нужной длины (easy) [0.2, 256]

Дано дерево из n вершин и $k \leq 10$ чисел L_i от 1 до $n-1$. Для каждого L_i посчитайте число простых путей в нашем дереве, состоящих ровно из k рёбер.

Формат входных данных

На первой строке число n ($1 \leq n \leq 10^5$), следующая строка содержит числа p_1, p_2, \dots, p_{n-1} . $0 \leq p_i < i$. p_i – отец вершины i в дереве. Далее k ($1 \leq k \leq 10$) и числа L_i .

Формат выходных данных

Выведите k чисел – количества соответствующих путей.

Примеры

stdin	stdout
5 0 1 1 3 2 3 2	2 4

Обязательные задачи

Задача 24С. Центроиды дерева [1.2, 256]

Дано дерево из n вершин. У каждой вершины есть цвет. Нужно обработать q запросов (v_i, c_i) : найти расстояние от v_i до ближайшей к v_i вершины цвета c_i . Расстоянием между вершинами называется минимальное количество рёбер в пути между ними.

Формат входных данных

На первой строке число n ($1 \leq n \leq 10^5$), следующая строка содержит числа p_1, p_2, \dots, p_{n-1} . $0 \leq p_i < i$. p_i – отец вершины i в дереве. Далее строка с числами a_0, a_1, \dots, a_{n-1} . $0 \leq a_i < n$. a_i – цвет вершины i . Далее строка с числом q ($1 \leq q \leq 10^5$). Следующие q строк содержат запросы $v_i q_i$ ($0 \leq v_i < n$, $0 \leq c_i < n$).

Формат выходных данных

Для каждого запроса выведите одно число – расстояние до ближайшей вершины нужного цвета, или -1 , если в дереве нет вершин такого цвета.

Примеры

stdin	stdout
5	0 1 2 -1 2 1 2 1 1
0 1 1 3	
1 2 3 2 1	
9	
0 1	
0 2	
0 3	
1 0	
2 1	
2 2	
3 3	
3 1	
4 2	

Задача 24D. БДБД [1, 256]

Большая Древесная База Данных создана для того, чтобы в ней можно было надежно сохранить и раскрасить любое дерево. В новой версии БДБД запланирован новый функционал, для реализации которого потребуется вновь переосмыслить теорию графов.

В БДБД хранится взвешенное дерево. В языке запросов Системы Управления Большой Древесной Базы Данных (СУБДБД) предусмотрены два вида запросов:

1. «1 v d c » — покрасить все вершины, находящиеся на расстоянии не более d от вершины v , в цвет c . Все вершины изначально окрашены в цвет с номером 0.
2. «2 v » — вывести цвет вершины v .

Необходимо запрограммировать работу СУБДБД и ответить на все запросы пользователя.

Формат входных данных

В первой строке число N ($1 \leq N \leq 10^5$) — количество вершин дерева.

Следующие $N - 1$ строк содержат описание ребер, по три числа в строке a_i, b_i, w_i ($1 \leq a_i, b_i \leq N, a_i \neq b_i, 1 \leq w_i \leq 10^4$), где i -ое ребро имеет вес w_i и соединяет вершины a_i и b_i .

В следующей строке число Q ($1 \leq Q \leq 10^5$) — число запросов. В каждой из Q следующих строк запросы одного из двух видов:

1. Числа 1, v, d, c ($1 \leq v \leq N, 0 \leq d \leq 10^9, 0 \leq c \leq 10^9$).
2. Числа 2, v ($1 \leq v \leq N$).

Все числа во входных данных целые.

Формат выходных данных

Для каждого запроса второго типа необходимо вывести в отдельной строке цвет запрошенной вершины.

Примеры

stdin	stdout
5	6
1 2 30	6
1 3 50	0
3 4 70	5
3 5 60	7
8	
1 3 72 6	
2 5	
1 4 60 5	
2 3	
2 2	
1 2 144 7	
2 4	
2 5	

Задача 24Е. Сбалансированные строки [0.6, 256]

Вам дано дерево из n вершин. Вершины пронумерованы целыми от 1 до n . Каждой вершине соответствует метка с символом '(' или ')'. Обозначим за $l[u \rightarrow v]$ строку, полученную конкатенацией всех меток на простом пути из u в v . Заметьте, для любой пары u и v есть ровно один простой путь.

Сбалансированные строки определяются следующим образом.

- Пустая строка сбалансирована.
- Для любой сбалансированной s , строка "(" s ")" сбалансирована.
- Для любых сбалансированных строк s и t , их конкатенация сбалансирована.
- Любая другая строка **не** сбалансирована.

Посчитайте количество таких пар вершин дерева (u, v) , что строка $l[u \rightarrow v]$ сбалансирована. Пары (u, v) и (v, u) считаются одинаковыми, смотрите первый пример.

Формат входных данных

На первой строке число вершин дерева n ($2 \leq n \leq 10^5$). Следующая строка содержит n символов '(' и ')'. i -й из них – метка i -й вершины дерева. Следующие $n - 1$ строк содержат рёбра дерева – пары чисел a_i и b_i ($1 \leq a_i, b_i \leq n$).

Формат выходных данных

Выведите одно число – ответ на задачу.

Примеры

stdin	stdout
2 (1 2	1
4 (() 1 2 2 3 3 4	2
5 (()) 1 2 2 3 2 4 1 5	4

Для искателей острых ощущений

Задача 24F. Строки в дереве [1.2, 256]

Дано дерево. Дерево — это связный граф без циклов. На каждом ребре дерева написана строчная латинская буква. Между каждыми двумя вершинами существует ровно один простой путь, то есть путь по рёбрам дерева, проходящий через каждую вершину не более одного раза. Каждому пути соответствует строка, которая получается, если идти по этому пути и читать буквы на рёбрах в порядке следования. Путь можно проходить, начиная с любого его конца.

Также дана строка S . Соответствует ли она какому-либо простому пути в данном дереве? Длина строки и размер дерева не превышают 300 000.

Вам необходимо реализовать функцию

`Solve (str, len, tree, n, v, u)`.

- `str` — строка, которую нужно найти
- `len` — длина строки
- `tree` — массив рёбер (`edge`)
- `n` — количество вершин в дереве
- `v, u` — переменные, в которые следует записать концы найденного вами пути

Функция должна возвращать 0 или 1 — присутствует ли строка в дереве.

Примеры

stdin	stdout
abc 4 1 2 c 4 3 a 2 4 b	YES 1 3
abc 1	NO
zy 6 1 2 x 1 3 y 1 4 z 3 5 a 4 6 b	YES 3 4

Пояснения к примерам

В первом примере строка `abc` соответствует пути 3–4–2–1.

Во втором примере в дереве нет ни одного ребра.

В третьем примере строка `zy` соответствует пути 3–1–4.

Для мастеров AI

Правила.

Это блок задач про промтинг. Пользоваться можно только бесплатными версиями ИИ. В шапке исходника указывать последовательность промтов, и сайты, куда они были отправлены. Если в процессе использования ИИ вы получили какие-то важные идеи для решения, это тоже часть решения. Если с помощью ИИ вы написали генератор тестов или стресс-тест, это тоже следует указывать. Задокументируйте, пожалуйста, проделанную работу в шапке отосланного решения. Если это нужно, вы можете писать часть кода сами.

Задача 24G. Приятная прогулка [1, 256]

В деревне Большой Костромич каждый вечер на прогулку выходят все мальчики, все девочки, а также все домашние животные (преимущественно собаки), живущие в деревне. Так оказалось, что в деревне одинаковое количество мальчиков, девочек и собак. При этом каждая девочка хочет обязательно гулять с каким-нибудь мальчиком, а мальчик — с собакой. В результате, все гуляющие жители деревни разбиваются на тройки, в каждой из которых оказывается мальчик, девочка и собака.

Для каждой двух жителей (включая собак) известно, сколько времени в течение прогулки они уделяют друг другу. Таким образом, прогулка мальчика, девочки и собаки протекает примерно так: сначала мальчик и собака играют в подвижные игры, не обращая никакого внимания на девочку, потом мальчик устает, и с собакой играет девочка, потом устает собака, и некоторое время мальчик с девочкой просто прогуливаются и болтают друг с другом. Когда им это надоедает, они расходятся. Таким образом, прогулка длится столько, сколько данный мальчик готов играть с данной собакой плюс сколько с данной собакой будет играть данная девочка плюс сколько эти мальчик и девочка готовы общаться друг с другом.

Требуется найти такое разбиение мальчиков, девочек и собак на тройки, чтобы среднее время прогулки в деревне было по возможности больше.

Формат входных данных

В первой строке входного файла записано число N — количество девочек в деревне ($1 \leq N \leq 100$). Далее записана матрица $N \times N$, j -ое число в i -ой строке этой матрицы задает, сколько времени мальчик номер i готов играть с собакой номер j . Далее записана еще одна матрица $N \times N$, j -ое число в i -ой строке этой матрицы задает, сколько времени собака номер i готова играть с девочкой номер j . Наконец, записана третья матрица $N \times N$, j -ое число в i -ой строке этой матрицы задает, сколько времени девочка номер i будет общаться с мальчиком номер j . Все числа в матрицах — натуральные, не превосходящие 10 000.

Формат выходных данных

В выходной файл выведите N троек чисел. Первое число каждой тройки должно задавать номер мальчика, второе — номер собаки, третье — номер девочки.

Замечание

Будут оцениваться программы, находящие решения не сильно хуже авторского. Вам нужно достигнуть качества 0.992 авторского.

Пример

stdin	stdout
3	2 1 3
5 1 4	3 2 2
3 4 6	1 3 1
5 6 3	
1 2 9	
3 5 3	
4 5 1	
2 9 3	
3 9 2	
4 8 3	

Подсказка по решению

Как оценить качество своего решения? Сгенерить случайные тесты, взять динамику за 2^{2n} . Чтобы получить ОК в системе, наверняка мелкие тесты придётся проходить динамикой.

Задача 24Н. Пути нужной длины (hard) [0.9, 256]

Дано дерево из n вершин. Для каждого k от 1 до $n-1$ посчитайте число простых путей в нашем дереве, состоящих ровно из k рёбер.

Формат входных данных

На первой строке число n ($1 \leq n \leq 10^5$), следующая строка содержит числа p_1, p_2, \dots, p_{n-1} .
 $0 \leq p_i < i$. p_i — отец вершины i в дереве.

Формат выходных данных

Выведите $n-1$ число.

Примеры

stdin	stdout
5 0 1 1 3	4 4 2 0