

## Содержание

<b>Must have</b>	2
Задача 10А. Самое дешевое ребро [0.5 sec, 512 mb]	2
Задача 10В. Задача про два станка ( $F2 \parallel C_{max}$ ) [0.2 sec, 512 mb]	3
<b>Обязательные задачи</b>	4
Задача 10С. Центроиды дерева [3.5 sec, 512 mb]	4
Задача 10D. БДБД [2.5 sec, 512 mb]	5
<b>Дополнительные задачи</b>	6
Задача 10Е. Гамильтонов путь конём [1 sec, 512 mb]	6
Задача 10F. Строки в дереве [2.5 sec, 512 mb]	7
Задача 10G. Arithmetic Encoding [0.2 sec, 512 mb]	8

---

Вы не умеете читать/выводить данные, открывать файлы? Воспользуйтесь **примерами**.

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) **переопределение стандартного аллокатора** ускорит вашу программу.

Обратите внимание на компилятор GNU C++11 5.1.0 (TDM-GCC-64) inc, который позволяет пользоваться **дополнительной библиотекой**. Под ним можно сдать **вот это**.

## Must have

### Задача 10А. Самое дешевое ребро [0.5 sec, 512 mb]

Дано подвешенное дерево с корнем в первой вершине. Все ребра имеют веса (стоимости). Вам нужно ответить на  $M$  запросов вида “найти у двух вершин минимум среди стоимостей ребер пути между ними”.

#### Формат входных данных

В первой строке файла записано одно числ —  $n$  (количество вершин).

В следующих  $n - 1$  строках записаны два числа —  $x$  и  $y$ . Число  $x$  на строке  $i$  означает, что  $x$  — предок вершины  $i$ ,  $y$  означает стоимость ребра.  $x < i, |y| \leq 10^6$ .

Далее  $m$  запросов вида  $(x, y)$  — найти минимум на пути из  $x$  в  $y$  ( $x \neq y$ ).

Ограничения:  $2 \leq n \leq 5 \cdot 10^4, 0 \leq m \leq 5 \cdot 10^4$ .

#### Формат выходных данных

$m$  ответов на запросы.

#### Пример

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

#### Замечание

Мы изучаем центроиды! Эта задача здесь, чтобы перед чем-то более сложным вы потренировались. Сдавайте только центроиды. Пойманные на баловстве получают минус баллы.

**Задача 10В. Задача про два станка ( $F2 \parallel C_{max}$ ) [0.2 sec, 512 mb]**

Имеется множество из  $n$  работ и два станка. Время выполнения  $i$ -й работы на первом станке равно  $a_i$ , время выполнения  $i$ -й работы на втором станке равно  $b_i$ . Каждую работу надо выполнить сначала на первом станке, потом на втором. И на первом, и на втором станке работы можно выполнять в произвольном порядке. Каждый станок в каждый момент времени может выполнять только одну работу.

Минимизируйте  $C_{max}$  — время выполнения последней работы на втором станке.

**Формат входных данных**

В первой строке дано одно целое число  $n$  ( $1 \leq n \leq 100\,000$ ) — количество работ.

В следующей строке  $n$  целых чисел от 0 до  $10^6$  — время выполнения работ на 1-м станке.

В следующей строке  $n$  целых чисел от 0 до  $10^6$  — время выполнения работ на 2-м станке.

**Формат выходных данных**

В первой строке выведите единственное число —  $C_{max}$ .

Во второй строке выведите перестановку — порядок выполнения работ на первом станке.

В третьей строке выведите перестановку — порядок выполнения работ на втором станке.

**Примеры**

stdin	stdout
3	16
1 2 3	1 3 2
5 5 5	1 2 3
2	6
3 2	2 1
1 3	2 1

## Обязательные задачи

### Задача 10С. Центроиды дерева [3.5 sec, 512 mb]

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  $v_i c_i$  ( $0 \leq v_i < n, 0 \leq c_i < n$ ).

#### 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 10D. БДБД [2.5 sec, 512 mb]

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  $v$ .

Запрограммируйте 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 тройки чисел  $a_i, b_i, w_i$  ( $1 \leq a_i, b_i \leq N, a_i \neq b_i, 1 \leq w_i \leq 10^4$ ),  $i$ -я строка обозначает ребро веса  $w_i$  между вершинами  $a_i$  и  $b_i$ .

В следующей строке число запросов  $Q$  ( $1 \leq Q \leq 10^5$ ).

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	

## Дополнительные задачи

### Задача 10E. Гамильтонов путь конём [1 sec, 512 mb]

Дана доска размера  $n \times n$  ( $5 \leq n \leq 400$ ).

Известно, что такую доску можно обойти конём, посетив каждую клетку ровно один раз.

#### Формат входных данных

Число  $n$ .

#### Формат выходных данных

Выведите гамильтонов **путь** –  $n \times n$  строк, каждая содержит очередную клетку.

Координаты нумеруются с нуля.

**Путь** можно строить из любой клетки. Если путей несколько, выведите любой.

#### Примеры

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

#### Замечание

Начинайте со случайной точки, и всё получится.

### Задача 10F. Строки в дереве [2.5 sec, 512 mb]

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.

### Задача 10G. Arithmetic Encoding [0.2 sec, 512 mb]

You have just obtained a job from Kitten Computing, the renowned software company. You cannot believe your luck – a lot of your friends have applied there before, but they either failed during the interview, or didn't manage to accomplish their first assignment and have been fired during their first month at Kitten Computing.

Now you've got your first assignment yourself, into a team that develops software for a new variant of arithmetic encoding. Recall that during arithmetic encoding the string to be packed is mapped into an interval  $(\alpha, \beta)$  of the real line, and then any fraction  $p/q$  lying on this interval is output.

Your exact task is to find this fraction  $p/q$ , with denominator and numerator as small as possible, for given rational numbers  $\alpha$  and  $\beta$ . You needn't bother about the origin of  $\alpha$  and  $\beta$ : the software for computing these numbers is being written by other programmers of your team.

#### Формат входных данных

Each line of the input file contains four integers  $P_1, Q_1, P_2, Q_2$ , such that  $\alpha = P_1/Q_1$  and  $\beta = P_2/Q_2$ . All numbers are non-negative and do not exceed  $10^{18}$ , the denominators  $Q_1$  and  $Q_2$  are non-zero, and it is guaranteed that  $\alpha < \beta$ . There will be at most  $10^4$  lines in the input.

#### Формат выходных данных

For each line of the input you are to output two positive integers,  $P$  and  $Q$ , separated by a single space, such that  $\alpha < P/Q < \beta$ , and  $Q$  is minimal possible. If there are several solutions, output one with minimal value of  $P$ .

#### Пример

stdin	stdout
3 10 7 20	1 3
0 1 1 1	1 2