Merge. Слияние последовательностей

Имя входного файла: `merge.in`
Имя выходного файла: `merge.out`

Даны две бесконечные неубывающие последовательности A и B. Требуется найти k-ый элемент в неубывающей последовательности C, содержащей все элементы из A и B (включая повторы).

Последовательность A задается с помощью полинома $P(x) = x^3$:

$$a_1 = P(1) \bmod 12345, \quad a_i = a_{i-1} + (P(i) \bmod 12345), \text{ при } i > 1$$

Последовательность B задается с помощью полинома $Q(x) = x^2$:

$$b_1 = Q(1) \bmod 123, \quad b_i = b_{i-1} + (Q(i) \bmod 123), \text{ при } i > 1$$

Формат входного файла

Входной файл содержит натуральное число k ($1 \le k \le 10^7$).

Формат выходного файла

В выходной файл выведите одно число — ответ на задачу. Гарантируется, что ответ не превышает $2 \cdot 10^9$.

Пример

merge.in	merge.out
1	1
3	5

Kth. K-ый минимум

Имя входного файла: `kth.in`
Имя выходного файла: `kth.out`

Напишите программу, которая находит k-ое в возрастающем порядке число в массиве $A = \langle a_1, a_2, \ldots, a_n \rangle$.

Массив A задается с помощью полинома $P(x) = 132x^3 + 77x^2 + 1345x + 1577$: $a_i = P(i) \bmod 1743$.

Формат входного файла

Входной файл содержит два натуральных числа n и k ($1 \le k \le n \le 50\,000$).

Формат выходного файла

В выходной файл выведите одно число — ответ на задачу.

Пример

kth.in	kth.out
1 1	1388
10 1	402

Sumdist. Сумма расстояний

Имя входного файла: `sumdist.in`
Имя выходного файла: `sumdist.out`

Дан связный граф. Требуется найти сумму расстояний между всеми парами вершин.

Формат входного файла

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($1 \le n \le 1\,000$, $0 \le m \le 10\,000$).

Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i — номерами концов ребра ($1 \le b_i, e_i \le n$).

Гарантируется, что граф связен.

Формат выходного файла

Первая строка выходного файла должна содержать одно натуральное число — сумму попарных расстояний между вершинами.

Пример

sumdist.in	sumdist.out
5 5 1 2 2 3 3 4 5 3 1 5	16

Longpath 1, 2. Длиннейший путь

Имя входного файла: `longpath.in`
Имя выходного файла: `longpath.out`

Дан ориентированный граф без циклов. Требуется найти в нем длиннейший путь.

Формат входного файла

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и дуг графа соответственно. Следующие m строк содержат описания дуг по одной на строке. Ребро номер i описывается двумя натуральными числами b_i и e_i — началом и концом дуги соответственно ($1 \le b_i, e_i \le n$).

Входной граф не содержит циклов и петель.

Варианты

Longpath1: $n \le 1\,000$, $m \le 10\,000$.
Longpath2: $n \le 10\,000$, $m \le 100\,000$.

Формат выходного файла

Первая строка выходного файла должна содержать одно натуральное число — количество дуг в длиннейшем пути.

Пример

longpath.in	longpath.out
5 5 1 2 2 3 3 4 3 5 1 5	3

Distance 1, 2, 3. Расстояние между вершинами

Имя входного файла: `distance.in`
Имя выходного файла: `distance.out`

Коль Дейкстру́ писать без кучи,
То тайм-лимит ты получишь...
А в совсем крутой задаче
Юзай кучу Фибоначчи!

Спектакль преподавателей ЛКШ.июль–2007

Дан взвешенный граф. Требуется найти вес минимального пути между двумя вершинами.

Формат входного файла

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно. Вторая строка входного файла содержит натуральные числа s и t — номера вершин, длину пути между которыми требуется найти ($1 \le s, t \le n$, $s \ne t$).

Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается тремя натуральными числами b_i, e_i и w_i — номера концов ребра и его вес соответственно ($1 \le b_i, e_i \le n$, $0 \le w_i \le 100\,000$).

Варианты

Distance1: $n \le 1\,000$, $m \le 10\,000$.
Distance2: $n \le 5\,000$, $m \le 100\,000$.
Distance3: $n \le 100\,000$, $m \le 200\,000$.

Формат выходного файла

Первая строка выходного файла должна содержать одно натуральное число — вес минимального пути между вершинами s и t.

Если путь из s в t не существует, выведите `-1`.

Пример

distance.in	distance.out
4 4 1 3 1 2 1 3 4 5 3 2 2 4 1 4	3

Spantree. Остовное дерево

Имя входного файла: `spantree.in`
Имя выходного файла: `spantree.out`

Требуется найти в связном графе остовное дерево минимального веса.

Формат входного файла

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно. Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается тремя натуральными числами b_i, e_i и w_i — номера концов ребра и его вес соответственно ($1 \le b_i, e_i \le n$, $0 \le w_i \le 100\,000$). $n \le 1\,000, m \le 10\,000$.

Граф является связным.

Варианты

Spantree1: $n \le 1\,000$, $m \le 10\,000$.
Spantree2: $n \le 5\,000$, $m \le 100\,000$.

Формат выходного файла

Первая строка выходного файла должна содержать одно натуральное число — вес минимального остовного дерева.

Примеры

spantree.in	spantree.out
4 4 1 2 1 2 3 2 3 4 5 4 1 4	7

Aplusb. Сложение

Имя входного файла: `aplusb.in`
Имя выходного файла: `aplusb.out`

Формат входного файла

Во входном файле два целых неотрицательных числа A и B ($A, B \leq 10^{10\,000}$), каждое в своей строке.

Формат выходного файла

В выходной файл выведите одно число, равное сумме A и B.

Пример

aplusb.in	aplusb.out
3 5	8