Задача A. Почти любимые числа

Автор задачи: Григорий Шовкопляс
Автор разбора: Григорий Шовкопляс

Требовалось найти количество чисел заканчивающихся числом n (почти любимых чисел), не превосходящих m.

Решение:

Разобьем число m на две части m_1 и m_2. То есть, если приписать к m_1 справа m_2, мы получим число m. При этом длина числа m_2 должна равняться длине числа n.

Ответ на задачу — число $m_1 + 1$, которое обозначает сколько существует вариантов для первых цифр с учетом ведущих нулей.

Есть только одно исключение, когда $m_2 < n$, тогда отпадает вариант почти любимого числа сформированного приписыванием к m_1 справа числа n, так как в этом случае оно будет больше m.

Для разбиения числа m на две части нужно было найти минимальное число x такое, что $10^x > n$, тогда $m_1 = m \div 10^x$, а $m_2 = m \mod 10^x$. ($\div$ — целочисленное деление, а $\mod$ — остаток).

Так как x не превосходит 9, решение работает быстро и легко укладывается в поставленные ограничения по времени.

Задача B. Пароль от сейфа

Автор задачи: Дмитрий Филиппов
Автор разбора: Дмитрий Филиппов

По данной строке требовалось сказать, можно ли поменять в ней местами не более чем два символа, чтобы она стала палиндромом.

Решение:

Для начала посчитаем количество «плохих» пар: таких $1 \le i < \frac{|s|}{2}$, что $s_i \ne s_{|s|-i-1}$.
Если количество таких пар больше 2, ответ «`NO`».
Если количество таких пар равно 0, ответ «`YES`».
Если пара одна, то есть два случая:

- Если длина строки четная, то ответ «`NO`».
- Если длина строки нечетная, надо проверить, что один из символов плохой пары равен центральному символу строки ($s_{|s|/2}$), иначе ответ «`NO`».

Если плохих пары две, надо разобрать несколько случаев и проверить, можно ли эти четыре символа как-то поменять местами, чтобы обе пары перестали быть плохими.

Задача C. Шифровка

Автор задачи: Дмитрий Филиппов
Автор разбора: Дмитрий Филиппов

По данному отрезку $[l, r]$ бесконечной строки 123456789101112... требовалось найти количество натуральных чисел, полностью лежащих в нем.

Решение:

Для начала найдем количество цифр в числе, через которое проходит левая граница отрезка. Это количество будет явно не больше 17, поэтому его можно было найти одним проходом цикла. Требовалось только вывести или заметить формулу количества цифр, требующихся для записи всех d-значных чисел: $9 \cdot 10^{d-1} \cdot d$.

После этого, зная, сколько в числе, которое пересекает левая граница отрезка, цифр, несложно найти и само число $lNumber$: $lNumber = \lceil (l - lSum - 1)/lDigits \rceil$, где $lDigits$ — количество цифр в числе, а $lSum$ — количество цифр, требующихся для записи всех $< lDigits$-значных чисел. Аналогичным образом решая задачу для числа r, находим: $rNumber = \lfloor (r - rSum)/rDigits \rfloor$.

Комментарий

Многие участники получали неправильные ответы на следующих тестах: $[101, 101]$, $[239, 239]$ из-за того, что выводили отрицательный ответ вместо 0.

Задача D. Захват провинций

Автор задачи: Илья Пересадин
Автор разбора: Илья Пересадин

Дано дерево и запросы: перекрасить вершину из белого в черный и наоборот и сказать, сколько на пути между двумя вершинами вершин из минимального связанного подграфа, покрывающего все черные вершины.

Решение:

Подвесим дерево за некоторую вершину. Посчитаем глубину каждой вершины. Заметим, что самая высокая вершина покрывающего подграфа — это lca двух самых дальних в порядке эйлерового обхода вершин. Обозначим эту вершину за v. Заметим также, что вершина принадлежит покрывающему подграфу, тогда и только тогда, когда она лежит в поддереве v и в ее поддереве содержится черная вершина.

Обойдем дерево dfs и запомним для каждой вершины u время входа и выхода для нее — $tin_u, tout_u$. Теперь каждой вершине u соответствует отрезок $[tin_u, tout_u]$, который задает ее поддерево, то есть для всех вершин t в ее поддерве выполнено $tin_u \le tin_t \le tout_u$.

Заведем структуру, которая будет помогать находить lca двух вершин. Например, посчитаем двоичные подъемы для каждой вершины. Будем поддерживать сет черных вершин, отсортированных по tin_u. При перекраске вершины будем удалять или добавлять из него перекрашенную вершину и пересчитывать lca двух самых дальних вершин в сете — вершину v. Также будем поддерживать дерево отрезков (или дерево Фенвика) для суммы и при перекраске вершины x увеличивать/уменьшать значение в точке tin_x на единицу.

На запрос второго типа для вершин x и y — найдем их lca и посчитаем количество вершин из покрывающего подграфа на пути от их lca до каждой и сложим — это и будет ответом на запрос.

Научимся считать количество вершин из покрывающего подграфа на пути от lca до x. Поднимемся от вершины x до ближайшей вершины из покрывающего подграфа — это можно сделать двоичным подъемом, проверяя для каждой вершины, что на отрезке вершины есть черная вершина, делая для этого запрос суммы к дереву отрезков. Теперь нужно рассмотреть случаи взаимного расположения вершины x, lca и v, и в зависимости от него посчитать количество вершин из покрывающего пографа на пути как разность высот двух из этих вершин. Разбор этих случаев предлагаем произвести самостоятельно.

Итого, каждый запрос первого типа обрабатывается за $log(n)$ операций, а запрос второго типа за $log^2(n)$ операций (двоичный подъем и запрос суммы).

Итоговое время $m \times log^2(n)$.

Итоговая память $n \times log(n)$.