

Сортировки. Разбор.

Задача А. Палиндром

Краткое условие: дано 10^5 символов, нужно составить из них палиндром максимальной длины, а из них - минимальный лексикографический.

Решение

Для двух различных палиндромов одинаковой длины, первая позиция, в которой они отличаются, находится не дальше их середины. То есть для сравнения палиндромов достаточно рассматривать их префикс до середины включительно. Отсортируем этот префикс и дополним его до палиндрома. Мы получили минимальный лексикографически палиндром.

Как с помощью этого теперь решать задачу: отсортировать символы в исходной строке и набирать символы по 2. Теперь, если мы не использовали все символы, нужно дополнить этот набор одним символом без пары. Если таких несколько выбрать из них минимальный.

Задача D. Число

Краткое условие: дано 100 строк состоящих из 100 цифр, нужно склеить их так, чтобы в итоге получилось максимально возможное число.

Решение

На первое место всегда лучше поставить как можно большую цифру. Среди таких первых кусочков выгоднее взять тот, у которого вторая цифра максимальна. И так далее.

По сути мы доказали, что нужно отсортировать имеющиеся числа по убыванию лексикографически (как строки), и поставить в таком порядке.

Важно понимать, что сортируем мы именно строки, потому что нам нужно, чтобы 9 было больше, чем, например, 80 ($9\ 80 > 80\ 9$).

Задача F. Коррозия металла

Краткое условие: есть N листов, i -ый лист выдерживает a_i времени под воздействием жидкости, находящейся в начале массива, и b_i времени под воздействием жидкости, находящейся в конце массива. Надо так поставить листы, чтобы максимизировать время, за которое жидкости встретятся.

Решение Достаточно отсортировать листы по величине $\frac{b_i}{a_i}$. Сложным моментом тут является посчитать ответ для такой сортировки: последний блок одновременно разъедается двумя жидкостями.

Сначала его часть отъест первая пришедшая жидкость, ее можно посчитать и вычесть. А потом две жидкости начнут раеждать вместе, скорости при это складываются:

$$T = \frac{1}{\frac{1}{a_i} + \frac{1}{b_i}} = \frac{a_i b_i}{a_i + b_i}$$

Как доказать, что это оптимально:

Давайте для удобства доказательства считать, что все b_i натуральные: если они рациональные, то их можно умножить на НОК их знаменателей, например, от умножения всех a_i и b_i на константу время не изменится. А вот если они иррациональные, то так, конечно, не получится, но в компьютере все числа в любом случае рациональные, так что примем это допущение.

Разобьем i -ый лист на a_i листов поменьше. Тогда каждый из этих единичных блоков будет разъедаться первой жидкостью за ровно 1 времени. При этом второй жидкостью они будут разъедаться за $\frac{b_i}{a_i}$ времени.

Заметим, что в такой формулировке очевидно, что все единичные блоки нужно располагать так, чтобы величина $\frac{b_i}{a_i}$ убывала: для первой жидкости порядок неважен, там везде время 1, а для торможения второй жидкости выгодно располагать блоки в порядке возрастания $\frac{b_i}{a_i}$.

Ну вот, осталось заметить, что когда мы располагаем сами большие блоки в порядке возрастания $\frac{b_i}{a_i}$, то и единичные блоки находятся в таком же порядке, а про него мы доказали, что он оптимальный, тем более он оптимальный и для больших блоков.

Задача G. Трубочист

Краткое условие: Есть N чисел в массиве, одно из них (K -й) зафиксировано, остальные можно переставлять как хочешь. Нужно переставить так, чтобы сумма модулей разностей соседних чисел была минимальной.

Решение

Рассмотрим 2 утверждения:

Утверждение 1:

Пусть разбиение чисел на множество лежащих слева от зафиксированного дома и справа от него зафиксировано. Тогда будет не хуже, если числа слева и справа отсортировать либо по возрастанию, либо по убыванию.

Утверждение 2:

Отсортируем все числа (кроме зафиксированного). Достаточно перебрать только случаи, когда первые $K - 1$ число - это подряд идущие числа, и случаи, когда последние $N - K$ чисел - это подряд идущие числа (в отсортированном массиве).

Из них сразу следует решение: отсортируйте все числа, кроме зафиксированного ($O(N \log N)$) после этого переберите все отрезки длины $K - 1$ и $N - K$ (это делается за $O(N)$). Для каждого разбиения нужно посчитать штраф, выбрав оптимально, в какую сторону отсортирована каждая половина (это делается для зафиксированного разбиения за $O(1)$).

Как доказывать Утверждение 1:

Посмотрим на числа слева от зафиксированного. Выкинем для начала все числа кроме трёх:

- минимальное из них a_{left_min}
- максимальное из них a_{left_max}
- зафиксированный дом a_k

Понятно, что $a_{left_max} - a_{left_min}$ всегда будет присутствовать в штрафе.

Также в ответе независимо всегда будет присутствовать такая разница:

- $|a_{left_min} - a_k|$, если $left_max < left_min < k$
- $|a_{left_max} - a_k|$, если $left_min < left_max < k$

А значит, штраф всегда хотя бы $a_{left_max} - a_{left_min} + \min(|a_{left_min} - a_k|, |a_{left_max} - a_k|)$.

Заметим, что в случае, когда все числа отсортированы, ровно такой штраф и будет. Сортировать при этом нужно так, чтобы этот минимум и достигался (по возрастанию, если $|a_{left_max} - a_k|$ меньше, и по убыванию, если $|a_{left_min} - a_k|$ меньше).

Как доказывать Утверждение 2:

Пусть все отсортированные незафиксированные числа - это $b_1 \leq b_2 \leq \dots \leq b_{N-1}$.

Рассмотрим минимальное и максимальное из чисел слева и справа:

- a_{left_min}
- a_{left_max}
- a_{right_min}
- a_{right_max}

Заметим, что среди них есть минимальное и максимальное число среди всех незафиксированных чисел b_1 и b_{N-1} . Пусть $b_1 = a_{left_min}$ (без ограничения общности). Тогда есть два случая:

Заметим, что тогда справа лежат числа между a_{right_min} и a_{right_max} . Давайте вместо них возьмем ровно $N - K$ чисел из отсортированного массива b , заканчивающихся на a_{right_max} , а что осталось запихнем в левую половину.

Если a_{right_max} - это не глобальный максимум (b_{N-1}), то штраф в левой половине никак не изменился, ведь она содержит и минимум b_1 , и максимум b_{N-1} . Если это глобальный максимум, то максимум левой половины после такой замены мог разве что уменьшиться до b_{K-1} , а уменьшение левого массива с максимальной из сторон не может увеличить штраф.

Правая половина стала с минимальной стороны только короче, а значит штраф тут тоже увеличиться не мог.

Оба утверждения доказаны.

Задача Н. Странные строки

Краткое условие: найти число различных странных подстрок в строке. Строка называется странной, если множество ее подстрок совпадает с множеством ее подпоследовательностей.

Решение: Докажем сначала такое утверждение, чтобы понять, что условие нас запутывает, и странная строка - это очень простая вещь:

Строка z является странной, если для нее выполняются следующие свойства:

- z содержит не более двух различных символов;
- все одинаковые символы в z идут подряд.

Действительно, пусть, например, в строке содержатся два одинаковых символа a , между которыми находится еще один символ, отличный от a . Пусть a встречается в строке k раз. Тогда строка из k символов a является подпоследовательностью z , но не ее подстрокой.

Пусть теперь в z встречаются хотя бы три различных символа. По доказанному, одинаковые символы должны идти подряд. Но тогда, если первый символ строки a , а последний - b , то строка ab является подпоследовательностью z , но не ее подстрокой. Наоборот, если строка z состоит из k символов a , после которых идет l символов b , то как $W(z)$, так и $Y(z)$ состоит из всех строк, в которых сначала идет не более k символов a , а затем не более l символов b .

Таким образом, для определения странности строки требуется найти количество ее подстрок, которые имеют такую структуру.

Будем решать задачу независимо для каждой упорядоченной пары (a, b) символов.

Пусть у нас есть несколько блоков подряд идущих символов a , после которых идет блок подряд идущих символов b . Сложим все пары (k_i, l_i) длин таких блоков в массив. Теперь нам надо найти число пар (k, l) , таких, что найдется (k_i, l_i) , такая что $1 \leq k \leq k_i$ и $1 \leq l \leq l_i$.

Для поиска числа таких пар отсортируем все пары по k_i , а затем по l_i . Для каждой пары (k_i, l_i) заменим l_i на максимум l_j , где $k_j \geq k_i$. Теперь оставим по одной паре с каждым k_i и просуммируем величины $(k_i - k_{i-1}) \times l_i$ для всех i . Получившееся значение и является искомым.

Отметим, что описанный алгоритм является ничем иным, как алгоритмом поиска площади объединения прямоугольников с осями, параллельными осям координат, общим углом $(0, 0)$ и положительными координатами противоположного угла.

В завершение описания полного решения отметим, что построение списков пар (k_i, l_i) можно осуществить одним проходом по массиву.