

Кубки

Авторы задачи: Александр Гордеев и Даниил Орешиников, разработчики: Владислав Власов и Даниил Орешиников

Упростим формулировку задачи: даны два массива различных целых чисел a и b . Требуется найти наименьший по длине массив c , содержащий два данных в качестве подпоследовательностей.

В первой подгруппе достаточно было написать полный перебор. Будем перебирать текущие позиции в a и в b и выбирать какой из элементов взять в ответ. Если очередные элементы a и b совпадают, то можно сдвинуть указатель в обеих. Такое решение работает за $O(2^{n+m})$.

Во второй подгруппе можно было воспользоваться динамическим программированием за $O(nm)$: будем искать $dp[i][j]$ — минимальный ответ, содержащий префикс a длины i и префикс b длины j как подпоследовательности. Для пересчета требовалось сравнивать длины длин двух строк и их первый символ. Переходы в динамике похожи на переходы при вычислении НОП двух последовательностей.

В исходной постановке задачи требовалось найти минимальную лексикографически строку, поэтому сравнение происходило за длину строки, и решение работало за $O(nm(n+m))$. В новой постановке такое решение работает за $O(nm)$ и также проходит пятую подзадачу.

Для полного посмотрим на алгоритм проверки наличия подпоследовательности в последовательности. Он просто итерируется по элементам большой последовательности, и когда встречает очередной элемент из подпоследовательности, переходит к следующему.

Тогда, если проверять наличие a и b как подпоследовательностей одновременно, то каждый очередной символ последовательности будет совпадать либо с очередным элементом a , либо с очередным элементом b , либо одновременно и с тем, и с тем. Длина итоговой большой последовательности, таким образом, равна $n+m-\text{common}$, где common — количество «общих» символов, встреченных на пути. Чтобы минимизировать итоговую длину, очевидно, что надо взять как можно больше общих символов, то есть длину наибольшей общей подпоследовательности a и b . То есть длина ответа будет равна $n+m-\text{lcs}(a,b)$.

В случае третьей подгруппы ответ восстанавливается несложно: достаточно одновременно идти по a и b двумя указателями и выписывать элементы в порядке возрастания, двигая соответствующий указатель. В таком случае и длина получившегося ответа будет минимальна, и среди всех таких ответов первый элемент будет минимальным.

В четвертой подгруппе все элементы b упорядочены по возрастанию, поэтому НОП a и b одновременно с этим будет возрастающей. Достаточно просто оставить только те элементы a , которые лежат и в b , и среди всех таких выбрать наибольшую возрастающую последовательность, что можно сделать за $O(n \log n)$. Остается восстановить ответ.

Понятно, что необходимо выписать все элементы a и b до их первого совпадающего элемента в любом порядке, затем совпадающий, затем все между первым и вторым совпадающим, и так далее. В качестве первого элемента будет выписан $\min(a_1, b_1)$, что является минимальным возможным, но только если $\min(a_1, b_1)$ не взят в качестве первого «общего» элемента. Чтобы избежать такой ситуации, из всех НОП выберем ту, которая начинается на максимальный возможный элемент. Это можно сделать, если запомнить состояния динамики «максимальная длина НВП, начинающейся в i -м элементе» для всех i .

Полное решение является обобщением предыдущей идеи. Все решение совпадает, но надо научиться находить НОП за $O(n \log n)$. Заметим, что раз все элементы a различны и все элементы b различны, можно для каждого a_i найти $\text{pos}[i]$ — индекс вхождения a_i в b , если такое вхождение есть. Тогда НОП a и b — это просто НВП массива pos .

Найдем НОП через НВП, после чего выберем из них ту, которая начинается на максимальный элемент, и восстановим ответ описанным выше способом.