Любитель камней

Автор задачи: Александр Шеховцов, разработчик задачи: Тимофей Ижицкий

В первой подгруппе можно предпосчитать для каждой подпоследовательности ее счет, и отвечать с помощью этого на запрос. Во второй подгруппе можно было написать динамику по строке s и t (примерно как в алгоритме поиска наибольшей общей подпоследовательности), но это не очень разумное решение, поэтому оно подробно не описывается.

В третьей подгруппе, ответ зависит только от длины строк — $\lfloor \frac{|s|-1}{|t|-1} \rfloor$.

Когда в задаче стоит вопрос о том, что нужно максимизировать какой-то минимум, часто приходит идея о бинарном поиске по ответу, давайте попробуем его сделать. Будем делать бинарный поиск по условию «правда ли, что есть подпоследовательность, что расстояние между соседними индексами хотя бы k» (где k — это значение, которое мы проверяем в бинарном поиске). Тогда можно такие индексы набирать жадно, будем идти жадно по индексам s и поочередно набирать текущий символ в t, каждый раз брать минимальный индекс хотя бы на k больше. Таким образом получили решение за $O(n \cdot \log T)$ на запрос, которое проходит 4 и 5 подгруппу.

Чтобы сдать задачу на 100 баллов нужно оптимизировать этот жадник, чтобы он быстрее работал на запрос. Поймем, что по сути в жаднике нужно находить для какой-то позиции pos в строке s, первую позицию $\geqslant pos$, что она равна очередному символу t_i. Давайте для каждой позиции и каждого символа из s это предпосчитаем, тогда в запросе также будем делать бинарный поиск и жадник, только двигаться по s с помощью предпосчитанного массива. Получили решение за $O(n \cdot \Sigma + T \cdot \log T)$, где Σ — размер алфавита (в данном случае 26).