

## Задачечка на подстрочечки

Будем обозначать  $count(h)$  равным числу подстрок строки  $h$ , которые встречаются в наборе  $s$ . Соответственно в запросах от нас требуется посчитать  $count(h[l_i, r_i])$ . Обозначим  $pref[i] = count(t[0 : i])$ , то есть число подстрок префикса строки  $t$  длины  $i$ , которые встречаются в наборе  $S$ . Обозначим  $suf[i] = count(t[i : |t|])$ , то есть число подстрок суффикса строки  $t$ , начиная с позиции  $i$ , которые встречаются в наборе  $S$ .

Тогда заметим, что  $pref[r] + suf[l] - count(t)$  равно  $count(t[l, r])$  минус число подстрок строки  $h$  из набора  $S$ , которые начинаются раньше позиции  $l$ , а заканчиваются позже позиции  $r$ . Заметим, что если подстрок второго типа нет, то ответ на запрос легко находится, все нужные значения  $pref[l]$  и  $suf[r]$  считаются с помощью Ахо-Корасика.

Если строки второго типа есть, то для запроса числа подстрок из  $S$  у  $t[l, r]$  найдем такую подстроку строки  $t$ , которая лежит в наборе  $S$ , и её вхождение в  $t$  начинается левее  $l$ , а заканчивается правее  $r$ . Среди всех таких строк найдем ту, у которой правая граница вхождения минимальная. Обозначим эту строку за  $s_i$ . Заметим, что строка  $t[l, r]$  входит в  $s_i$  как подстрока, обозначим это за  $s_i[l', r']$ . Так же заметим, что в  $s_i$  нет подстрок, которые начинаются левее  $l'$ , заканчиваются правее  $r'$  и входят в  $S$  как подстрока (иначе для подстроки  $t[l, r]$  нашлась бы более близкая справа строка, которая её покрывает. А значит, можно применить исходные рассуждения с префиксами и суффиксами для строки  $s_i$  и найти ответ.

Чтобы для запроса  $t[l, r]$  найти нужную строку  $s_i$ , пройдемся сканирующей прямой по строке  $t$ . С помощью Ахо-Корасика для каждой позиции  $i$  найдем максимальную подстроку  $t$  из набора  $S$ , которая заканчивается в позиции  $i$  (пусть это строка  $s_j$ ). Также в структуре куча или set будем хранить начала всех запросов, у которых концы не позже позиции  $i$ . Тогда для любого запроса, начинающегося позже начала вхождения  $s_j$ , строка  $s_j$  будет подходящей. Все такие запросы далее удаляются из кучи.

Таким образом, итоговое решение потребует построения Ахо-Корасика для прямых и развернутых строк, а так же прохода по строке  $t$  с поддержкой запросов в куче. Итоговая асимптотика:  $O(S + |t| + m \log m)$ .