

1. Найдите количество различных подстрок в строке s за $O(n \log n)$, где $n = |s|$.
2. Найдите наибольшую общую подстроку строк s и t за $O(n \log n)$, где $n = \max\{|s|, |t|\}$.
3. Заданы строки s и t . Для каждого $k \in [0, |t|]$ найдите максимальное $\ell = \ell(k)$, такое что префикс строки s длины ℓ входит в t хотя бы k раз. Асимптотика: $O(n \log n)$, где $n = \max\{|s|, |t|\}$.
4. Задан текст t с известным суффиксным массивом. По строке s определите количество её вхождений в t за время $O(|s| \cdot \log |t|)$.
5. Найдите максимальное количество непересекающихся вхождений строки s в текст t за $O(n \log n)$, где $n = \max\{|s|, |t|\}$.
6. В строке s найдите наибольшую строку, которая входит без пересечений в s хотя бы k раз, за $O(n \log^2 n)$, где $n = |s|$.
7. Дан массив чисел a_1, \dots, a_n . Два его подотрезка $[x, y]$ и $[u, v]$ назовём похожими, если выполнены все три условия: а) отрезки не пересекаются; б) они имеют одинаковую длину; в) $\forall i \in [0, y - x]$ выполнено $a_{x+i} + a_{u+i} = a_x + a_u$. Для каждого из q отрезков $[x_i, y_i]$ найдите количество похожих на него. Асимптотика: $O((n + q) \log^2 n)$.

1. Постройте суффиксный массив на строке s и массив lcp на нём. Проходите по суффиксам в порядке сортировки. Сколько новых подстрок вносит каждый такой суффикс? Воспользуйтесь тем, что множество всех подстрок совпадает со множеством префиксов всех суффиксов.
2. Постройте суффиксный массив на строке $t\#s$ и массив lcp на нём. Для поиска наибольшей общей подстроки достаточно рассмотреть соседние суффиксы в этом порядке (проверьте, что они пришли из разных строк).
3. Постройте суффиксный массив на строке $t\#s$ и массив lcp на нём. Найдите вхождение строки s в него. Для каждого ℓ (в порядке, скажем, убывания) найдите, на сколько можно отступить от этого вхождения вниз и вверх так, чтобы все пройденные суффиксы начинались на $s_0s_1 \dots s_{\ell-1}$?
4. Для каждого очередного символа s запустите два бинарных поиска, чтобы найти отрезок вхождений соответствующего префикса s в суффиксный массив строки t .
5. Найдите все вхождения s в t , а затем набирайте их жадно, чтобы получать непересекающиеся вхождения.
6. Для каждого ℓ разбейте весь суффиксный массив на блоки, внутри каждого из которых все суффиксы имеют общий префикс длины хотя бы ℓ . В каждом из них работает жадный алгоритм. Осталось найти оптимальное ℓ бинарным поиском.
7. Условие в) равносильно такому: $\Delta_{x+i} = -\Delta_{u+i}$, где Δ — массив разностей соседних элементов. Запишите в один массив Δ и $-\Delta$, разделённые $\#$. Далее, найдём вхождение суффикса $a_{x_i} \dots$ в суффиксный массив. Интересует подотрезок, содержащий этот суффикс, на котором $\text{LCP} \geq y_i - x_i$. На этом отрезке остаётся найти такие отрезки, которые не пересекаются с $[x_i, y_i]$. Поможет дерево отрезков, отвечающее на запрос “количество чисел, меньше либо равных x , на данном отрезке”.