

Всюду в этом листке считаем, что алфавит — константного размера.

1. Постройте суффиксное дерево для строки *abaabab*\$ алгоритмом Укконена.
2. Покажите, как построить суффиксный массив и массив `lcp` по суффиксному дереву за $O(n)$, где n — длина строки.
3. Покажите, как построить суффиксное дерево по суффиксному массиву за $O(n)$, где n — длина строки (суффиксный массив строить не нужно).
4. В строке s найдите количество различных её подстрок за $O(n)$.
5. Дан словарь из слов s_1, \dots, s_n со стоимостями c_1, \dots, c_n . Для произвольной строки t определим $f(t) = |t| \cdot \sum_{i=1}^n c_i \cdot \text{cnt}(t, s_i)$, где $\text{cnt}(u, v)$ — количество вхождений u в v . Найдите максимальное значение $f(t)$ по всем строкам t за время $O(L \log n)$, где $L = \sum_{i=1}^n |s_i|$.
6. В строке s найдите k -ю (с учётом числа вхождений) подстроку в лексикографическом порядке.
7. По данной строке научитесь за $O(1)$ отвечать на вопросы равенства двух подстрок.
8. В данной строке найдите самую длинную подстроку, входящую в неё хотя бы два раза без пересечений.

1. Для проверки ответа обратитесь к лекции.
2. Достаточно запустить dfs-обход дерева, сначала проходя по меньшим (в алфавитном порядке) буквам. Наибольший общий префикс двух строк соответствует самой глубокой их общей вершине.
3. Добавляйте суффиксы в порядке их следования в суффиксном массиве. Поднимайтесь вверх, исходя из значения lcp .
4. Каждая подстрока s однозначно соответствует пути из корня суффиксного дерева.
5. Создайте строку, составленную из s_1, \dots, s_n с попарно различными разделителями (размер алфавита станет $\Theta(n)$). Перебирайте t как пути из корня (достаточно рассматривать вершины). Достаточно просуммировать стоимости строк, соответствующих достижимым разделителям из поддеревя.
6. Постройте суффиксное дерево. Для каждой строки-вершины найдите число вхождений в строку (обратите внимание на число листьев в поддереве). Затем для каждой вершины найдите суммарное число вхождений подстрок, начинающихся с этой вершины. Теперь задача уже классическая: найти в боре k -ю строку.
7. Рассмотрите два соответствующих суффикса и найдите их LCP. Это их LCA в суффиксном дереве.
8. Для каждой вершины-строки можно легко узнать её первое и последнее вхождения: достаточно найти самый длинный путь до листа и самый короткий.