Алгоритмы. Теор.листок 2

Артемий Клячин

26 октября 2021 г.

Задача 1.

Построим общее сжатое суффиксное дерево sufftree для строк S, S^{-1} за $O(n)$.

$LCP(S,T)$ - наибольшая общий префикс строк S и T.

$LCA(S,T)$ - наименьший общий предок вершин соответствующих строкам S и T.

Пусть $LCE(S,T,i,j) = LCP(S[i:], P[j:])$

$LCE(S,T,i,j)$ определяется меткой LCA на sufftree для $S[i:]$ и $T[j:]$. Т.е. для листов суффиксов $S[i:]$ и $T[j:]$ надо найти ближайшего общего предка, а по глубине предка узнает длину наибольшего общего префикса.

Есть способ находить LCA за $O(1)$ с препроцессингом за $O(N)$ (алгоритм Фарах-Колтона и Бендера). Таким образом мы можем делать LCE за $O(1)$.

$LCE(S, S^{-1}, i, (n-1)-i)$ - это длина наибольшего палиндрома нечётной длины с центром в i.

$LCE(S, S^{-1}, i+1, (n-1)-i)$ - это длина наибольшего палиндрома чётной длины с центром в i и $i+1$.

Таким образом за $O(n)$ мы получим ответ на задачу.

Итоговая асимптотика $O(n)$.

Задача 2.

Введём понятия:

Строка y *сильно* продолжает строку x, если после каждого вхождения x в s следует вхождение y.

Строка y *очень слабо* продолжает строку x, если y слабо и несильно продолжает строку x.

Построим сжатое суффиксное дерево.

Возьмём произвольный x - ему соответствует какая-то позиция на каком-то ребре.

Количество сильных продолжений - это расстояние в символах до ближайшей вершины дерева. (далее будет разветвление и сильного продолжения не существует)

Далее посчитает очень слабые продолжения. Если мы подсчитаем их для вершины v, то их количество будет таким же для всех позиций на ребре (p, v) (кроме самой вершины p), где p - вершина-предок вершины v.

Так как любые два очень слабых продолжения x либо совпадают, либо один является префиксом другого, то очень слабые продолжения находятся на одном пути в некоторый лист u.

Другие продолжения не могут быть такой же длины, что и очень слабое продолжение, следовательно, посимвольная глубина позиции, в которую попадает очень слабое продолжение должна быть больше, чем у любого другого продолжения не на пути к u.

Любое подобное продолжение на пути к u будет очень слабым.

Следовательно, количество очень слабых продолжений x будет $h_1 - h_2$, где h_1 и h_2 - длины двух самых глубоких невложенных продолжения ($h_1 \geq h_2$).

Алгоритм:
за $O(N)$ строим дерево, за $O(N)$ при помощи DFS ищем 2 самых больших продолжения для каждой вершины, за $O(N)$ обходим все ребра и для каждого x, за $O(1)$ вычисляем количество слабых (сильных и очень слабых) продолжений.

e - ребро сжатого суффиксного дерева.
$len(e)$ - количество x на ребре e.
v_e - более глубокая вершина e
Ответ: $\sum\limits_{e} \frac{len(e) \cdot (len(e)-1)}{2} + m \cdot (h_1(v_e) - h_2(v_e))$

Задача 3.

Построить суффиксное дерево $s_1\$_1 s_1\$_2 ... s_n\$_n$, где $\$_1, ..., \$_n$ - разные окончания.

Далее каждой строкой пройдём по дереву. Если для строки s_i есть продолжение отличное от $\$_i$, то для s_i ответ YES, иначе NO.

Оценка асимптотики:

1) Построение дерева - $O(\sum\limits_{i=1}^{n} |s_i|)$

2) Проход всех строк по дереву происходит суммарно за $O(\sum\limits_{i=1}^{n} |s_i|)$.

Итоговая асимптотика: $O(\sum\limits_{i=1}^{n} |s_i|)$

Задача 4.

Построим общее сжатое суффиксное дерево для $s_1, s_1, ..., s_n$. В каждой вершине дерева будем хранить список строк, для которых данная вершина является терминальной. Такие списки будем называть списками терминальных состояний.

Далее каждой строкой s_i пройдёмся по дереву и обработаем для каждой вершины списки терминальных состояний. Если для строки s_j, вершина, в которую мы попали - терминальная, то существует суффикс s_j, который является префиксом s_i. И наоборот, если p строка является префиксом s_i и суффиксом s_j, то она обязательно попадётся при обходе. Так как нас интересуют максимальные такие p, то будем искать последнее вхождение строки s_j в списки терминальных состояний.

По итогу мы получили по $n-1$ значений для каждой строки s_i. Которые и будут ответом на задачу.

Оценка асимптотики:

1) Построение дерева - $O(n)$

2) Пусть $m = \sum\limits_{i=1}^{n} |s_i|$.

Для каждой строки производим спуск вниз по дереву и обрабатываем терминальные списки. Спуск производится за $O(m)$. Для каждой строки s_i, количество терминальных вершин $|s_i|$, следовательно, длины всех терминальных списков m, поэтому суммарно для всех $s[i]$ их обработка займёт $O(m)$.

3) По итогу алгоритма имеется n списков по $n-1$ значению. Ответ выводим за $O(n^2)$

Итоговая асимптотика: $O(\sum\limits_{i=1}^{n} |s_i| + n^2)$

Задача 5.

Количество различных подстрок в строке - это сумма длин всех рёбер в сжатом суффиксном дереве +1 (пустое подслово).

Решим задачу с помощью алгоритма Укконена. Алгоритм Укконена постепенно строит дерево: в начале строит дерево для $s[0,0]$, потом $s[0,1]$, ... $s[0,n-1]$. Дополнительно, на каждой итерации, считаем сумму длин рёбер суфф.дерева для $s[0,i]$, используя результат для предыдущей итерации: $i-1$

Так как асимптотика Укконена $O(|s|)$, то асимптотика алгоритма $O(|s|)$.