Применения суффиксных структур

Гусев Илья

Московский физико-технический институт

Москва, 2018

Содержание

Нахождение числа уникальных подстрок

1. Построение суфф. массива
2. Построение массива LCP

$$\sum_{i=0}^{n-1} |S_{Suf[i]}| - LCP[i]$$

Наибольшая общая подстрока K строчек

Суфф. массив

1. Построение суфф. массива для $concat(S^1, \$^1 \dots S^k, \$^k)$
2. Построение массива LCP
3. Поддерживаем окошко от i до j (индексы в суфф.массе), в котором встретились подстроки из всех K строк
4. Минимум LCP на этом окошке - общая подстрока K строк
5. Окошко двигаем, меняем максимум
6. Сложность?

Наибольшая общая подстрока K строчек

Суфф. массив

1. Построение суфф. массива для $concat(S^1, \$^1 \dots S^k, \$^k)$
2. Построение массива LCP
3. Поддерживаем окошко от i до j (индексы в суфф.массе), в котором встретились подстроки из всех K строк
4. Минимум LCP на этом окошке - общая подстрока K строк
5. Окошко двигаем, меняем максимум
6. Сложность: $O(n \cdot log(n))$ или $O(n)$ на построение суфф. массива, $O(n)$ на Касаи, $O(n)$ на основную часть

Наибольшая общая подстрока K строчек

Суфф. дерево

1. Построение обобщённого суфф. дерева для K строк
2. Для внутренних вершин пишем, откуда мы в них приходили. Например, если мы в нееё дошли из 1, 2, 3 строки, пишем {1, 2, 3}
3. Наиболее глубокая вершина с $\{1, \ldots, k\}$ - ответ
4. Сложность?

Наибольшая общая подстрока K строчек

Суфф. дерево

1. Построение обобщённого суфф. дерева для K строк
2. Для внутренних вершин пишем, откуда мы в них приходили. Например, если мы в нееё дошли из 1, 2, 3 строки, пишем {1, 2, 3}
3. Наиболее глубокая вершина с {1, . . . , K} - ответ
4. Сложность: $O(n)$ на Укконена, $O(n \cdot K)$ на основную часть. Можно ли за $O(n)$? Можно!

Наибольшая общая подстрока K строчек

Суфф. дерево

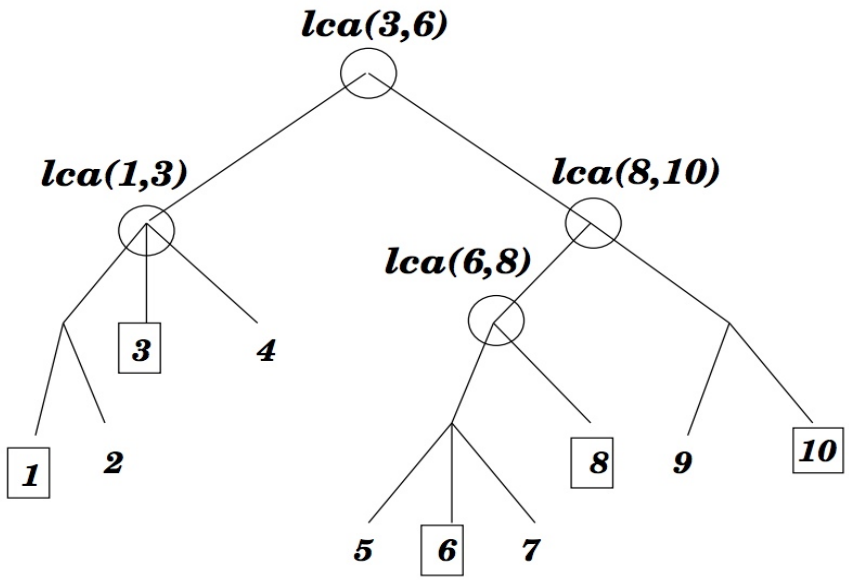
1. Построение обобщённого суфф. дерева для K строк
2. Нам достаточно количества уникальных встреч со строками $S^1 \dots S^k$!
3. Можем посчитать количество листьев в любом поддереве за O(n) Обозначим за $S(v)$ количество листьев в поддереве вершины v
4. Можем посчитать количество 'дубликатов' для i-ой строки в любом поддереве за O(n)! Обозначим это за U(v)
5. Для каждой вершины узнаём C(v) = S(v) - U(v). Где C(v) == k и наибольшая глубина - наш ответ
6. Сложность: $O(n)$

Наибольшая общая подстрока K строчек

Суфф. дерево

Считаем количество 'дубликатов' для i-ой строки в любом поддереве за O(n).

1. DFS, запоминаем номера суффиксов i-ой строки
2. Считаем LCA для соседних номеров в получившихся массивах
3. Для каждой вершины, считаем сколько раз мы считали LCA в поддереве
4. Это и есть U(v)



Наибольшая подстрока-палиндром

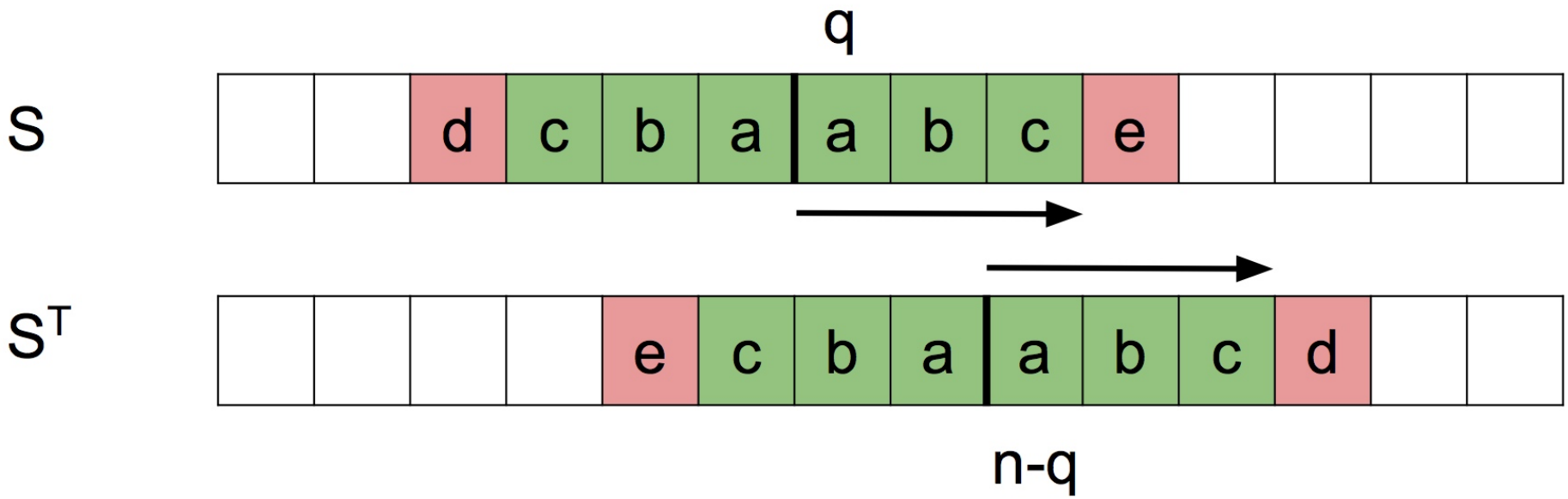
Суфф. массив

1. $SR = s_1 \dots s_n \# s_n \dots s_1 \$$
2. Построение суфф. массива
3. Построение LCP
4. Идеём последовательно по массиву
   - Если левый суффикс из прямой строки, а правый из обратной (или наоборот), то это нам подходит.
   - Кроме того, проверяем, что суффикс обратной строки захватывает суффикс прямой (их пересечение непустое, lcp + s + r = len(S))
   - Берём максимум из LCP таких пар

Наибольшая подстрока-палиндром

Суфф. дерево

1. $S = s_1 \dots s_n\#$
2. $S^T = s_n \dots s_1\$$
3. Рассматриваем S_i и S^T_{n-i+1}
4. Ищем их LCA, смотрим на его глубину
5. Повтрояем для всех i, берём вершину с наибольшей глубиной
6. Достраиваем её до палиндрома в зависимости от чётности



Поиск количества непересекающихся вхождений строки в текст

Суфф. массив

- Построение суфф. массива
- Построение LCP
- Поиск вхождений строки в текст (непрерывный диапазон в суффиксном массиве)
- Сортировка этого диапазона по оригинальному номеру суффикса
- Магия индексов

Полезные ссылки I

- APL6: Common substrings of more than two strings
  http://web.cs.ucdavis.edu/ğusfield/cs224f11/commonsubstrings.pdf
- SO: Longest palindrome in a string using suffix tree
  https://bit.ly/2A7Q5BX
- Quora: How can I find the longest common substring of three or more strings using a suffix array?
  https://bit.ly/2yz80PX