Алгоритм Касаи

Гусев Илья

Московский физико-технический институт

Москва, 2018

Содержание

Задача

Вычислить длину наибольших общих префиксов (LCP) для всех соседних суффиксов строки, отсортированных в лексикографическом порядке

Алгоритм Касаи, Аримуры, Арикавы, Ли, Парка

Str	a	a	b	a	a	c	a	#
Idx	0	1	2	3	4	5	6	7
Suf	7	6	0	3	1	4	2	5
0	#	a	a	a	a	a	b	c
1		#	a	a	b	c	a	a
2			b	c	a	a	a	#
3			a	a	a	#	c	
4			a	#	c		a	
5			c		a		#	
6			a		#			
7			#					
LCP	#	0	1	2	1	1	0	0

Утверждение 1: $\forall x < y \leq z, LCP(S_{Suf[y-1]}, S_{Suf[y]}) \geq LCP(S_{Suf[x]}, S_{Suf[z]})$
Пример: $\mathrm{LCP}(S_3, S_0) \geq LCP(S_4, S_6)$

Алгоритм Касаи, Аримуры, Арикавы, Ли, Парка

Str	a	a	b	a	a	c	a	#
Idx	0	1	2	3	4	5	6	7
Suf	7	6	0	3	1	4	2	5
0	#	a	a	a	a	a	b	c
1		#	a	a	b	c	a	a
2			b	c	a	a	a	#
3			a	a	a	#	c	
4			a	#	c		a	
5			c		a		#	
6			a		#			
7			#					
LCP	#	0	1	2	1	1	0	0

Утверждение 2: $\mathrm{LCP}(S_{Suf[x-1]}, S_{Suf[x]}) > 1 \Rightarrow$
$S_{Suf[x-1]+1} < S_{Suf[x]+1}, Suf^{-1}[Suf[x-1]+1] < Suf-1[Suf[x]+1]$
Пример: $x = 3$, $\mathrm{LCP}(S_0, S_3) = 2 \Rightarrow S_1 < S_4, 4 < 5$

Алгоритм Касаи, Аримуры, Арикавы, Ли, Парка

Str	a	a	b	a	a	c	a	#
Idx	0	1	2	3	4	5	6	7
Suf	7	6	0	3	1	4	2	5
0	#	a	a	a	a	a	b	c
1		#	a	a	b	c	a	a
2			b	c	a	a	a	#
3			a	a	a	#	c	
4			a	#	c		a	
5			c		a		#	
6			a		#			
7			#					
LCP	#	0	1	2	1	1	0	0

Утверждение 3: $\mathrm{LCP}(S_{Suf[x-1]}, S_{Suf[x]}) > 1 \Rightarrow LCP(S_{Suf[x-1]+1}, S_{Suf[x]+1}) = LCP(S_{Suf[x-1]}, S_{Suf[x]}) - 1$

Пример: $x = 3$, $\mathrm{LCP}(S_0, S_3) = 2 \Rightarrow LCP(S_1, S_4) = LCP(S_0, S_3) - 1 = 1$

Алгоритм Касаи, Аримуры, Арикавы, Ли, Парка

$p = Suf^{-1}[i-1], q = Suf^{-1}[i], j-1 = Suf[p-1], k = Suf[q-1]$
S_{j-1} - сосед слева S_{i-1} в суфф.массе, S_k - сосед слева S_i в суфф. массе
Теорема: LCP$(S_{j-1}, S_{i-1}) > 1 \Rightarrow LCP(S_k, S_i) \geq LCP(S_{j-1}, S_{i-1}) + 1$.

1. $Suf^{-1}[j] < Suf^{-1}[i]$
2. $Suf^{-1}[j] \leq Suf^{-1}[k] = Suf^{-1}[i] - 1 \Rightarrow LCP(S_k, S_i) \geq LCP(S_j, S_i)$
3. $LCP(S_j, S_i) = LCP(S_{j-1}, S_{i-1}) - 1$

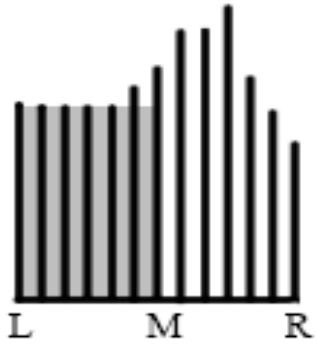
Итерации: $S_1 \ldots S_n$.На каждой итерации текущее значение LCP может быть не более чем на единицу меньше предыдущего. $\Rightarrow O(2n)$

Поиск подстроки за $O(log|S| + |pattern|)$

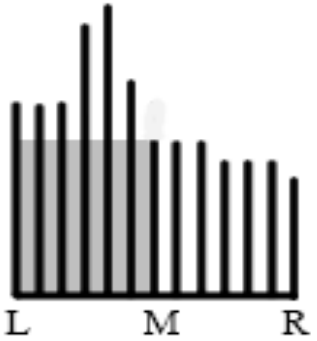
1. L — левая граница текущего диапазона поиска (изначально равна 0),
2. R — правая граница текущего диапазона поиска (изначально равна |S|1),
3. M=(L+R)/2 — середина текущего диапазона поиска,
4. $l = LCP(S_L, p)$ — длина общего префикса образца и левого края текущего диапазона поиска,
5. $r = LCP(S_R, p)$ — длина общего префикса образца и правого края текущего диапазона поиска,
6. $m_l = LCP(S_L, S_M)$ — длина общего префикса середины текущего диапазона и левого края текущего диапазона поиска,
7. $m_r = LCP(S_R, S_M)$ — длина общего префикса середины текущего диапазона и правого края текущего диапазона поиска.

Поиск подстроки за $O(log|S| + |pattern|)$

(1)



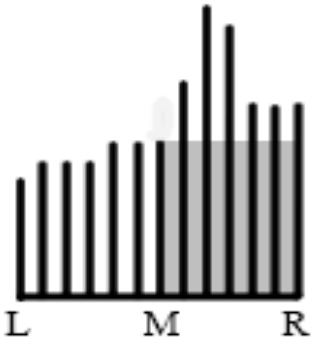
(2)



(3)

(1)

(2)



(3)

Нахождение числа уникальных подстрок

1. Построение суфф. массива
2. Построение массива LCP

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

Полезные ссылки I

- Викиконспекты: алгоритм Касаи
  https://bit.ly/2ygcpHb
- Викиконспекты: алгоритм поиска подстроки в строке с помощью суффиксного массива
  https://bit.ly/2OahBqT
- Emaxx: суффиксный массив
  http://e-maxx.ru/algo/suffix_array