

# Algoritmi i strukture podataka

5.čas

19. novembar 2020.

## 1 Rešenja

i	SA[i]	sufiks	LCP[i]
1	12		0
2	1	abbadabbado	5
3	6	abbado	1
4	4	adabbado	2
5	9	ado	0
6	3	badabbado	3
7	8	bado	1
8	2	bbadabbado	4
9	7	bbado	0
10	5	dabbado	1
11	10	do	0
12	11	o	0
13	0	yabbadabbado	

1. Sufiksni niz date niske je 12, 1, 6, 4, 9, 3, 8, 2, 7, 5, 10, 11, 0.  
Sufiksni niz neke niske  $s$  je sortirani niz svih sufiksa niske  $s$ .

Algoritam Karkainen-Sanders se sastoji iz 3 etape:

1. sortiranje sufiksa koji počinju na pozicijama oblika  $3^*k+1$  i  $3^*k+2$
2. sortiranje sufiksa koji počinju na pozicijama oblika  $3^*k$
3. učešljavanje (mrdžovanje) ova dva niza

Najpre ćemo dodati dve 0 na kraj niza, tj. na pozicije  $s[s.length]$  i  $s[s.length+1]$ .  
0 označava prazan karakter i tu je samo da bismo pokrili neke granične slučajeve. Označimo sa  $S_i$  sufiks koji počinje na poziciji  $i$  u datom stringu  $s$ . Sortiranje u fazi 1. radimo koristeći radix sort. Produkt ove faze je sortirani niz sufiksa koji počinju na pozicijama oblika  $3^*k+1$  i  $3^*k+2$ .

U našem slučaju dobijamo:

$$S_1 < S_4 < S_8 < S_2 < S_7 < S_5 < S_3 < S_{10} < S_{11}.$$

Sortiranje u fazi 2. vršimo oslanjajući se na već sortirane sufikse iz faze 1. Sufiks  $S_i$  možemo predstaviti u obliku  $S_i = s[i]S_{i+1}$ . Npr.  $S_7 = bS_8$ ,  $S_{10} = dS_{11}$ .

U našem slučaju važi:

$$S_0 = yS_1, S_3 = bS_4, S_6 = aS_7, S_9 = aS_{10}, S_{12} = 0S_{13}.$$

Upoređivanje u ovoj fazi se sada svodi na poredjenje prvog karaktera i pozicije sufiksa  $S_{i+1}$  u sortiranom nizu pod 1.

$S_0 = (y; 1)$  (jer se sufiks  $S_1$  nalazi na poziciji 1 u sortiranom nizu dobijenom pod 1),  $S_3 = (b; 2)$ ,  $S_6 = (a; 5)$ ,  $S_9 = (a; 7)$ ,  $S_{12} = (0; 0)$ .

Dobijamo  $(0; 0) < (a; 5) < (a; 7) < (b; 2) < (y; 1)$ , tj.

$$S_{12} < S_6 < S_9 < S_3 < S_0.$$

U fazi 3. koristimo sličnu ideju kao u fazi 2. Rastavimo sufikse koristeći jednakosti  $S_i = s[i]S_{i+1}$  i  $S_i = s[i]s[i+1]S_{i+2}$  i uporedimo ih na osnovu početnih karaktera i pozicije  $S_i$  i  $S_{i+1}$  u nizu pod 1.

2. LCP niz ove niske je 0, 5, 1, 2, 0, 3, 1, 4, 0, 1, 0, 0.

LCP[i] (longest common prefix) je dužina najdužeg zajedničkog prefiksa dva uzastopna sufiksa niske u sortiranom redosledu (sufiksa na pozicijama  $i$  i  $i+1$  u sufiksnom nizu).

3. U prilogu je slika.

U prilogu imate i urađen primer izracunavanja sufiksnog niza i sufiksnog stabla za nisku cattcat.

4. \* Rešenje koje je obrađeno na predavanju je dobijeno korišćenjem sufiksnog niza.

Rešenje koje koristi sufiksno stablo je <https://www.geeksforgeeks.org/suffix-tree-application-2-searching-all-patterns/?ref=lbp>

5. \* Rešenje koje je obrađeno na predavanju je dobijeno korišćenjem sufiksnog niza i niza LCP.

Rešenje koje koristi sufiksno stablo je <https://www.geeksforgeeks.org/suffix-tree-application-5-longest-common-substring-2/?ref=lbp>

6. Rešenje koje koristi sufiksni niz predstavlja modifikaciju algoritma iz zadatka 5.

Rešenje koje koristi sufiksno stablo je <https://www.geeksforgeeks.org/suffix-tree-application-6-longest-palindromic-substring/>

7. Rešenje koje koristi sufiksno stablo je <https://www.geeksforgeeks.org/suffix-tree-application-3-longest-repeated-substring/?ref=1bp>