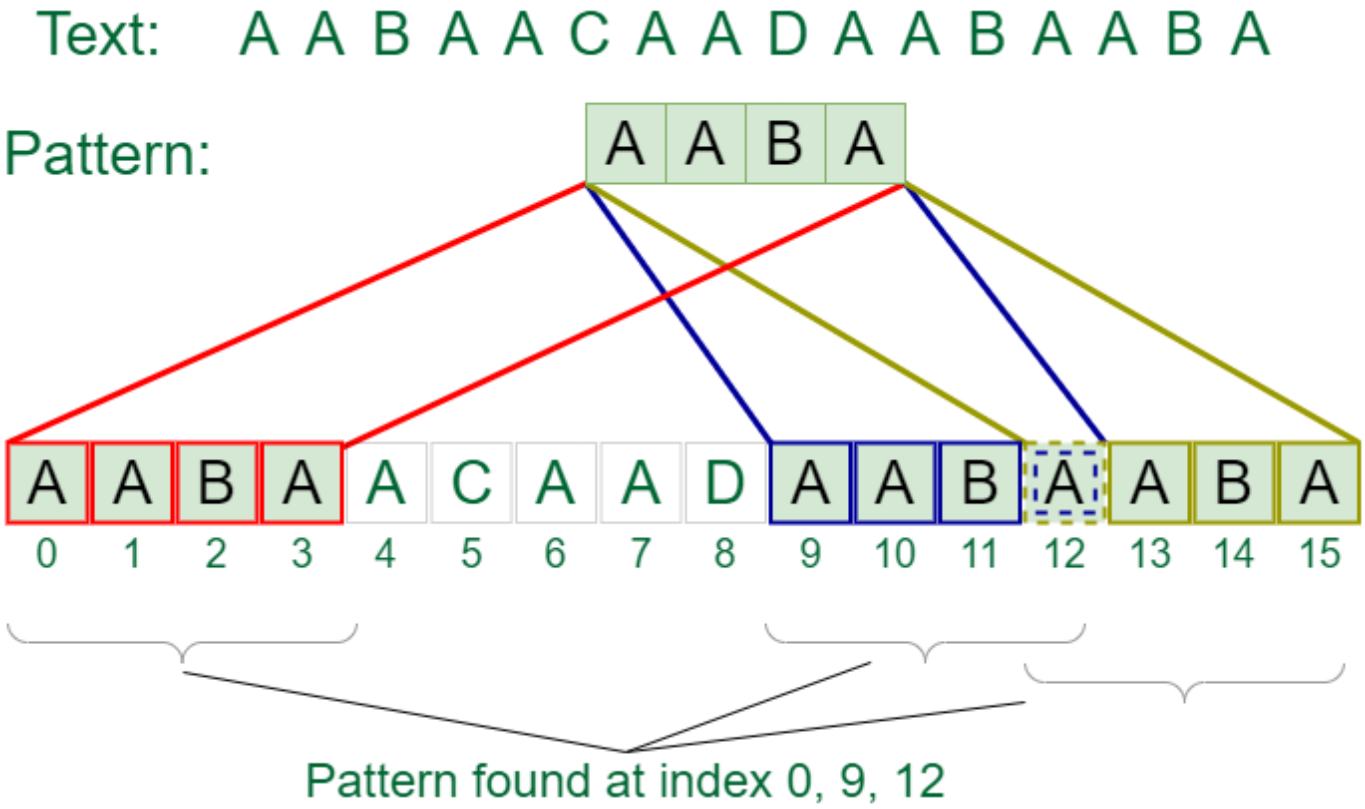
Algoritmi konstrukcije sufiksnog niza

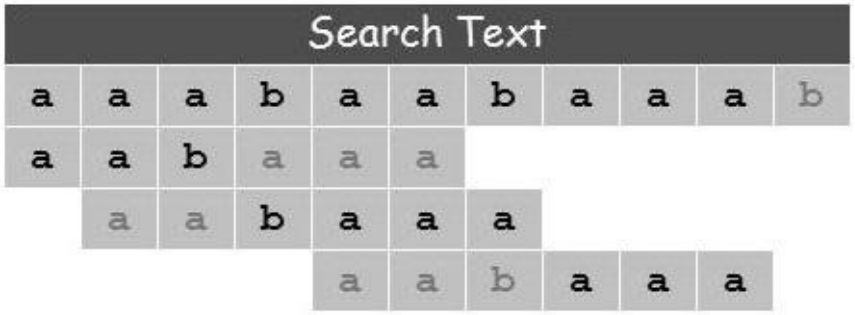
Lazar Vasović, 2006/2021
prof. dr Miodrag Živković
Algoritmi teksta – napredni koncepti
Seminar Katedre, 15. septembar 2022.

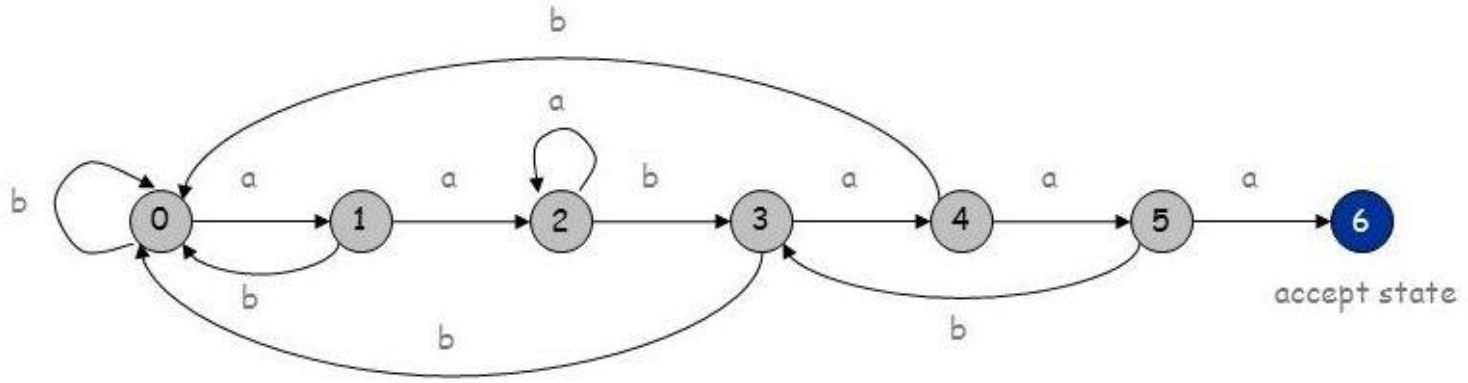
Pretraživanje niske



- Traženje jedne niske (šablona) u drugoj (bazi) – osnovni problem koji rešavaju algoritmi teksta

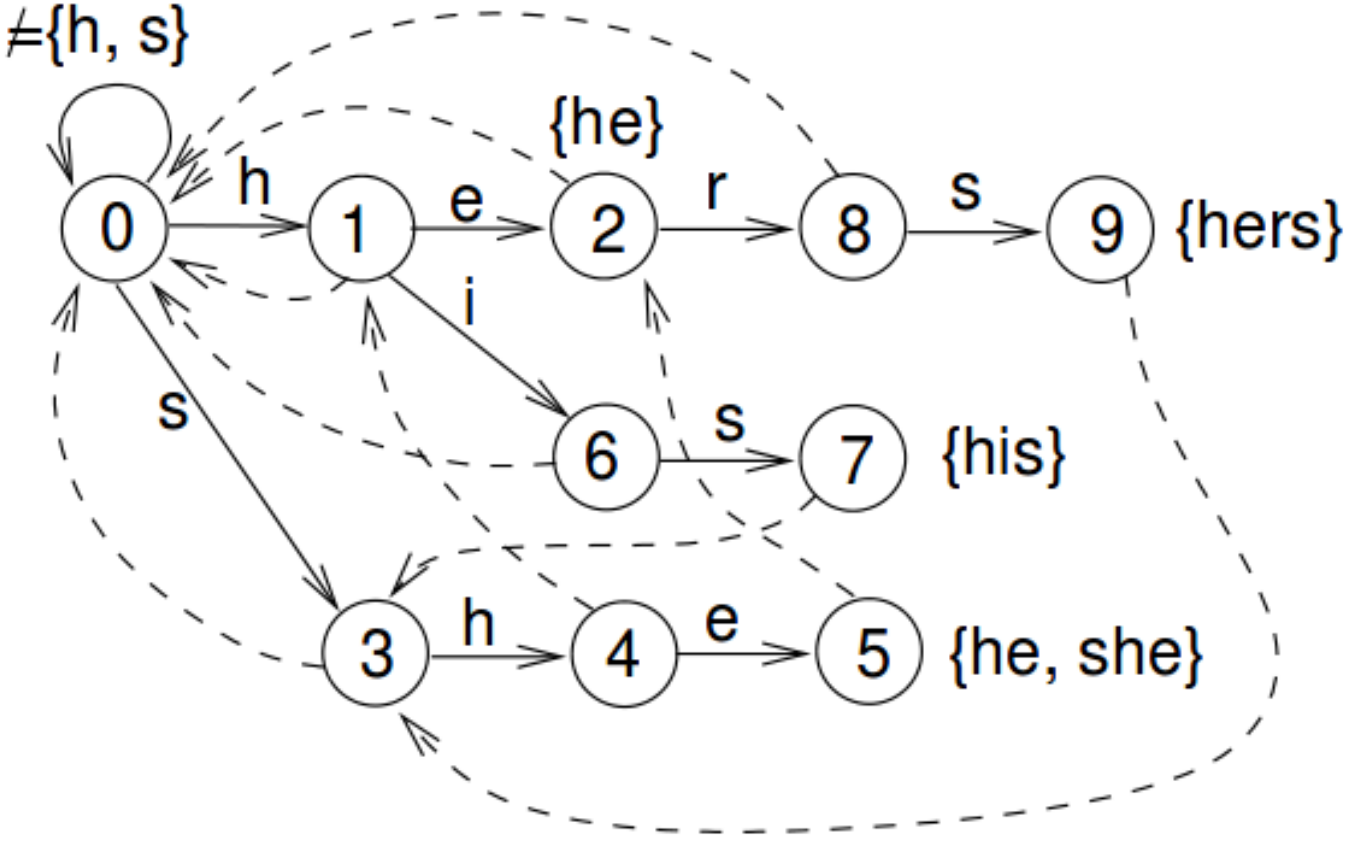
Priprema jednog šablona





- Tradicionalni pristup pripremi jednog šablona – Knut-Moris-Prat i odgovarajući konačni automat

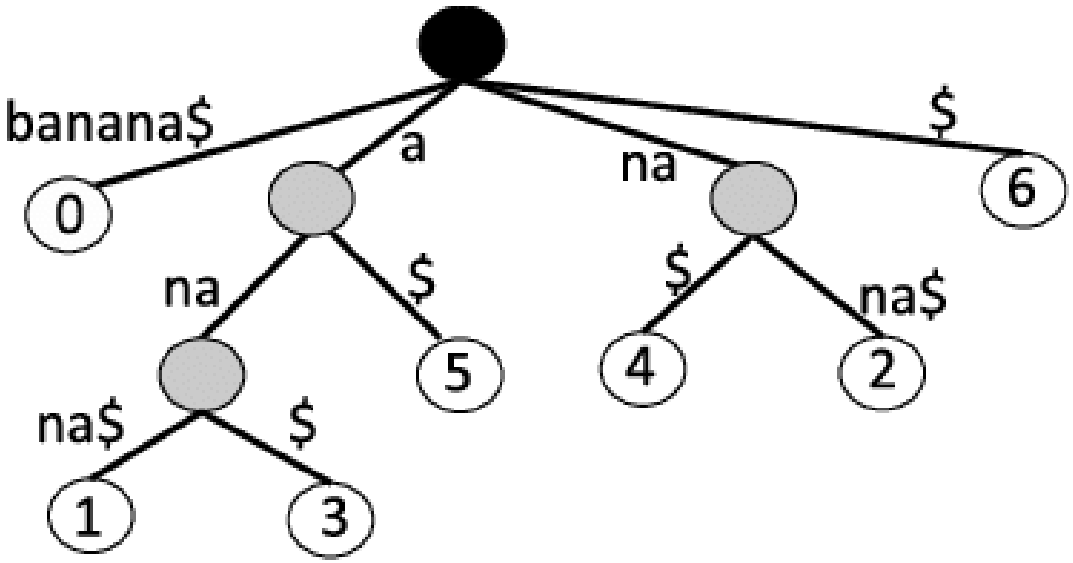
Priprema više šablona



- Tradicionalni pristup pripremi više šablona – Ejho-Korasik i odgovarajuće prefiksno stablo

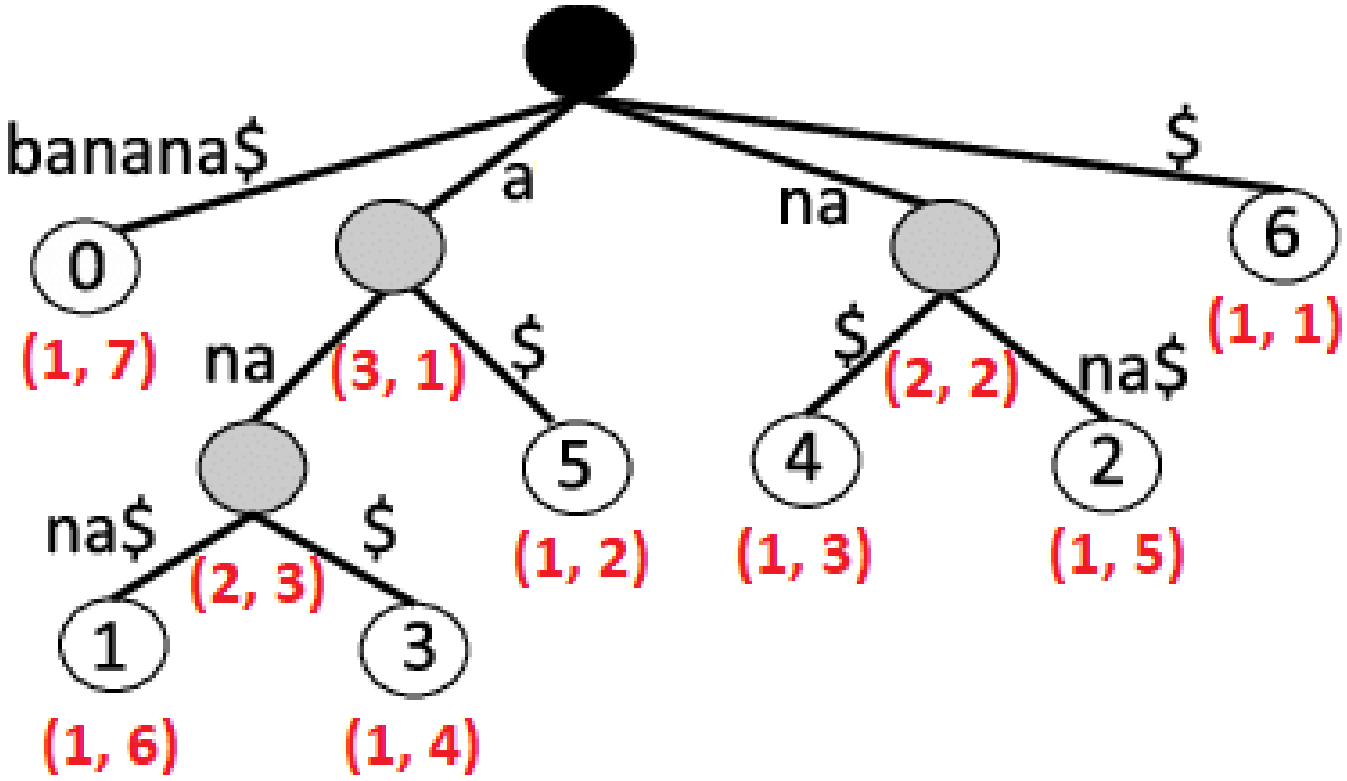
Priprema baze

0	1	2	3	4	5	6
b	a	n	a	n	a	$



- Šabloni su promenljivi – uglavnom je isplativija predobrada baze
- Priprema bazne niske – sufiksno stablo (prefiksno stablo sufiksa)

Sufiksno stablo



- Sufiksno stablo – dobro opisuje internu strukturu niske
- Predobrada baze – efikasno rešavanje mnogih zadataka

Sufiksni niz

Idx	Suffixes
0	BANANA
1	ANANA
2	NANA
3	ANA
4	NA
5	A

SA-Idx	Idx	Sorted Suffix
0	5	A
1	3	ANA
2	1	ANANA
3	0	BANANA
4	4	NA
5	2	NANA

Suffix Array [5, 3, 1, 0, 4, 2]

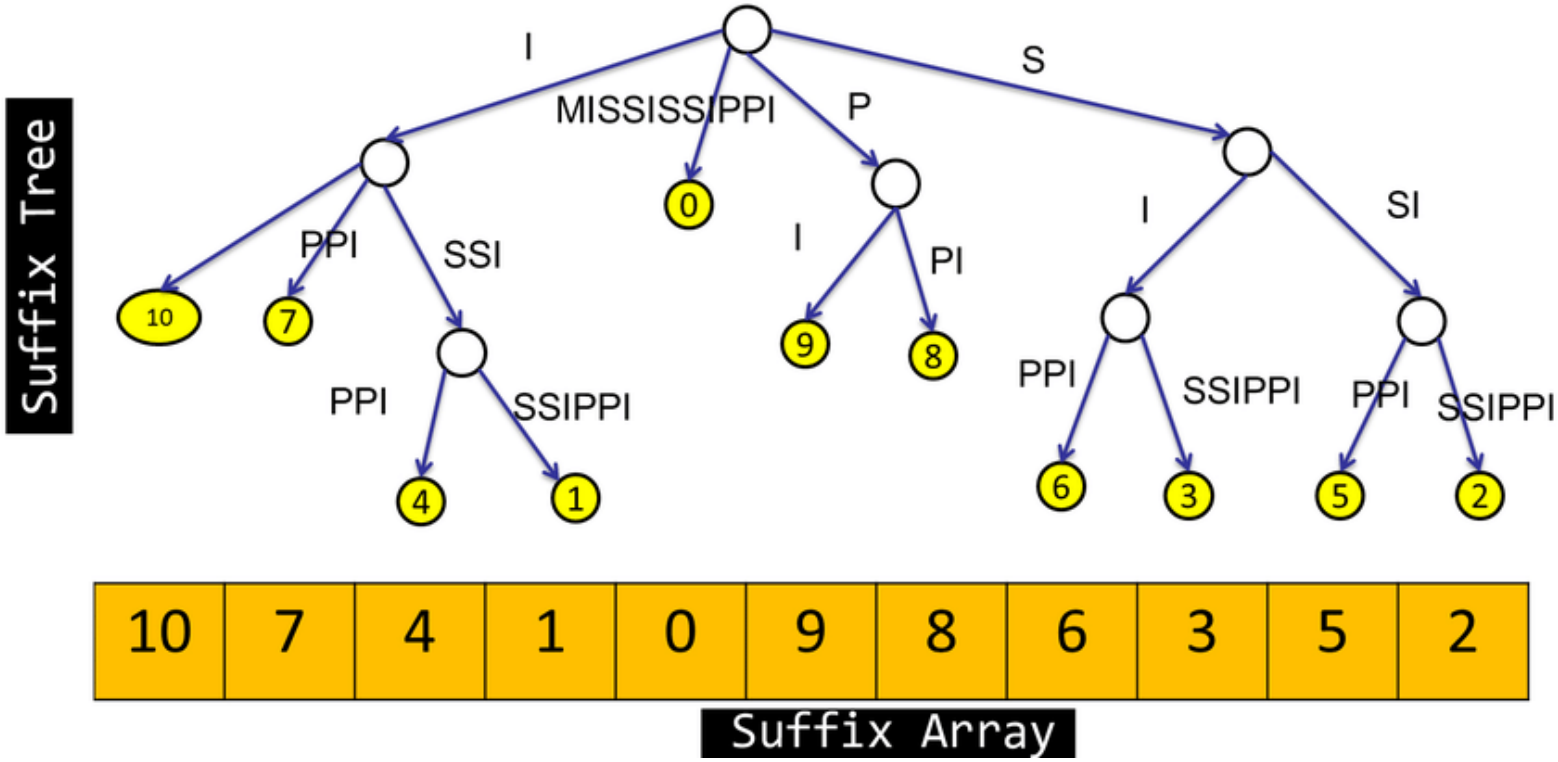
- Sufiksno stablo – korisno, ali prostorno zahtevno i složeno
- Sufiksni niz – leksikografski sortirani niz indeksa sufiksa

Najduži zajednički prefiksi

Suffix	Index	LCP
A	5	0
ANA	3	1
ANANA	1	3
BANANA	0	0
NA	4	0
NANA	2	2

- Najduži zajednički prefiksi – pomoćni niz
- Svi problemi – efikasna linearna rešenja ✓

Niz iz stabla



- Prost obilazak sufiksnog stabla leksikografskim poretkom
- Vremenski efikasno, ali suštinski ne rešava problem prostora

Naivni algoritam

- Prosto sortiranje indeksa prema sufiksima

Str	m	i	s	s	i	s	s	i	p	p	i
Idx	0	1	2	3	4	5	6	7	8	9	10
Niz	10	7	4	1	0	9	8	6	3	5	2
Rang	5	4	11	9	3	10	8	2	7	6	1

- Efikasno sortiranje – $O(n\log n)$ poređenja
- Poređenje sufiksa – $O(n)$ zbog više karaktera
- Sveukupna vremenska složenost – $O(n^2 \log n)$
- Prostorna složenost – u mestu (jedini)

Dupliranje prefiksa

- Sufiksni nisu nezavisni – potiču iz iste niske
- Mogu se iterativno sortirati, prema prefiksima
- Zapravo se rangiraju – $R_{2k}[i] = rang(R_k[i], R_k[i+k])$
- U *j*-toj iteraciji poredak prema 2^j karaktera

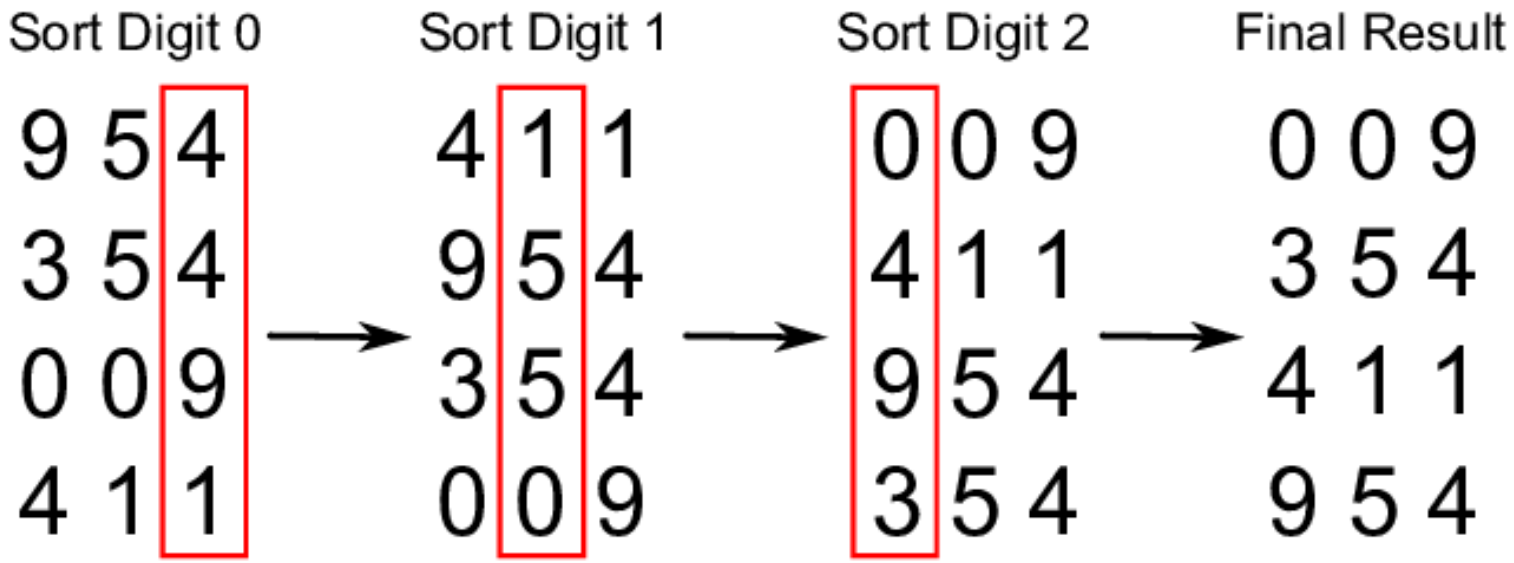
S_1	m	i	s	s	i	s	s	i	p	p	i
S_2	m-i	i-s	s-s	s-i	i-s	s-s	s-i	i-p	p-p	p-i	i
S_4	mi-ss	is-si	ss-is	si-ss	is-si	ss-ip	si-pp	ip-pi	pp-i	pi	i
S_8	miss-issi	issi-ssip	ssis-sipp	siss-ippi	issi-ppi	ssip-pi	sipp-i	ippi	ppi	pi	i
S_{16}	missisi-ppi	ississip-pi	ssissipp-i	sissippi	issippi	ssippi	sippi	ippi	ppi	pi	i

Primer dupliranja i analiza

S_0	m	i	s	s	i	s	s	i	p	p	i
R_1	2	1	4	4	1	4	4	1	3	3	1
S_1	(2, 1)	(1, 4)	(4, 4)	(4, 1)	(1, 4)	(4, 4)	(4, 1)	(1, 3)	(3, 3)	(3, 1)	(1, 0)
R_2	4	3	8	7	3	8	7	2	6	5	1
S_2	(4, 8)	(3, 7)	(8, 3)	(7, 8)	(3, 7)	(8, 2)	(7, 6)	(2, 5)	(6, 1)	(5, 0)	(1, 0)
R_4	4	3	10	8	3	9	7	2	6	5	1
S_4	(4, 3)	(3, 9)	(a, 7)	(8, 2)	(3, 6)	(9, 5)	(7, 1)	(2, 0)	(6, 0)	(5, 0)	(1, 0)
R_8	5	4	11	9	3	10	8	2	7	6	1
Niz	10	7	4	1	0	9	8	6	3	5	2

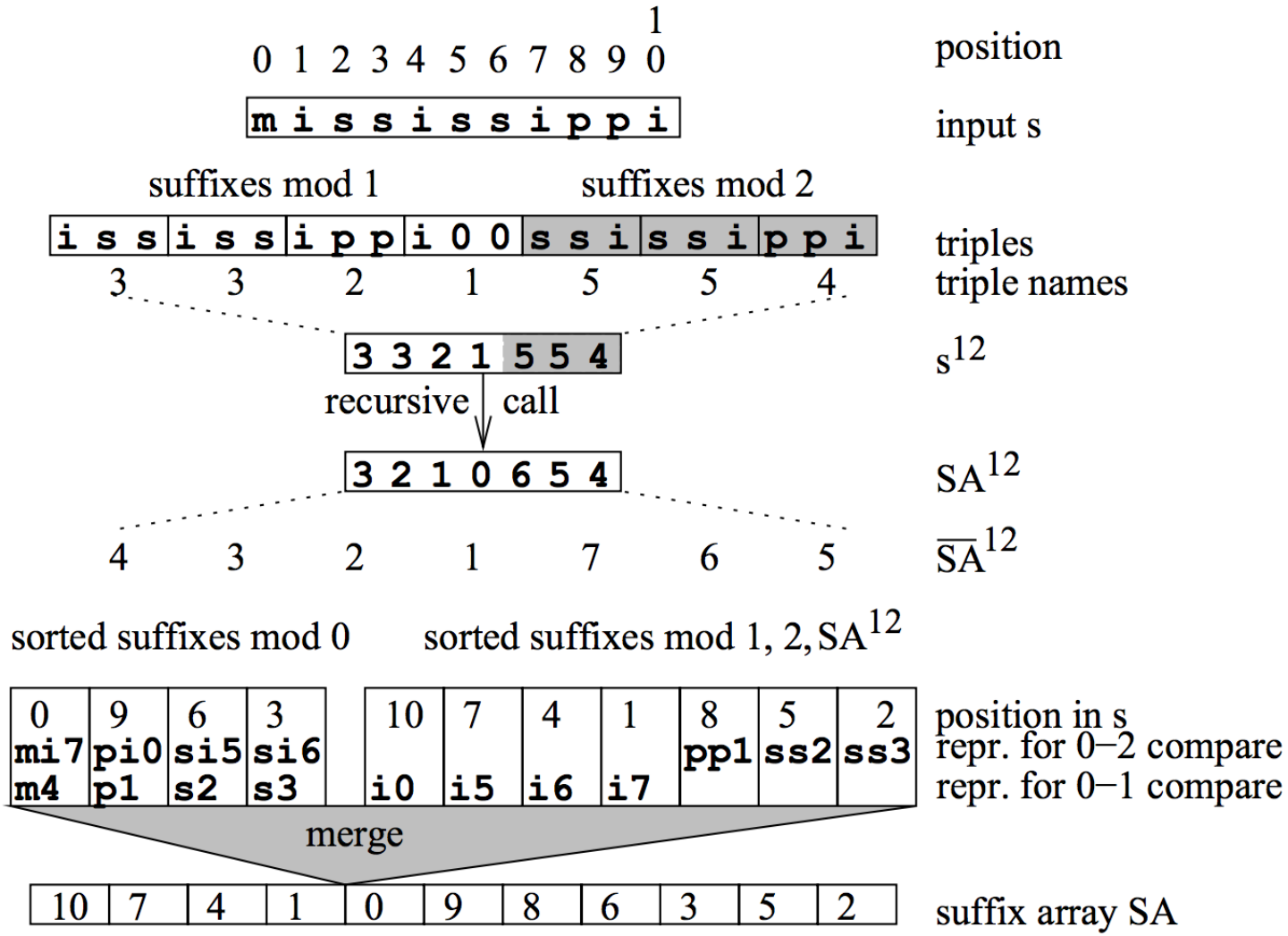
- Broj sortiranja – $O(\log n)$, zbog dupliranja
- Efikasno sortiranje – $O(n \log n)$ poređenja
- Sveukupna vremenska složenost – $O(n \log^2 n)$

Sortiranje razvrstavanjem



- Torke fiksnog opsega – sortiranje bez poređenja
- Razvrstavanje – $O(kn)$ tj. $O(n)$ jer je fiksno $k = 2$
- Sveukupna vremenska složenost – $O(n \log n)$

Algoritam *DC3*



Analiza algoritma *DC3*

- Razvrstavanje, rangiranje, spajanje – $O(n)$
- Rekurzivni poziv – rangovi dve trećine veličine
- Sveukupno – $T(n) = T(\frac{2}{3}n) + O(n)$, $T(n) = O(n)$
- Jedan od prvih algoritama linearne vremenske složenosti – uveden 2003, dopunjen 2006.
- Autori – Kerkejnen (*Kärkkäinen*) i Sanders
- Naziv – pokrivač/nje razlike po modulu 3 (*difference cover modulo 3*), zbog podele

Korektnost algoritama 🤔

- Algoritmi su efikasni, ali da li stvarno rade ⚠️
- Sami algoritmi – autori su pokazali ispravnost
- Implementacije – napisana svita testova
- Poznata rešenja – *banana*, *mississippi*
- Specijalno – prazna niska, niska dužine jedan
- Slučajne niske – poređenje sa naivnim
- Svi testovi prolaze – visoka sigurnost ✔️

Zaključak

- Urađeno – opisana, realizovana i upoređena četiri algoritma konstrukcije sufiksnog niza
- Rezultati – ...

Literatura

- Mohamed Ibrahim Abouelhoda, Stefan Kurtz, Enno Ohlebusch. Replacing suffix trees with enhanced suffix arrays. *Journal of Discrete Algorithms*. 2(1):53–86, 2004.
- Juha Kärkkäinen, Peter Sanders, Stefan Burkhardt. Linear work suffix array construction. *Journal of the ACM*. 53(6):918–936, 2006.
- Ge Nong, Sen Zhang, Wai Hong Chan. Two Efficient Algorithms for Linear Time Suffix Array Construction. *IEEE Transactions on Computers*. 60(10):1471–1484, 2011.