

1. Mājasdarbs

Lietišķajos Algoritmos (DatZ4020)

Kaspars Neiders

Stud. Apl. Nr. KN16012

29.09.2019

Lietišķie algoritmi – 1. mājas darbs

Termiņš: 2019. gada 30. septembris. Laiks 23:59:59 Eastern European Summer Time.

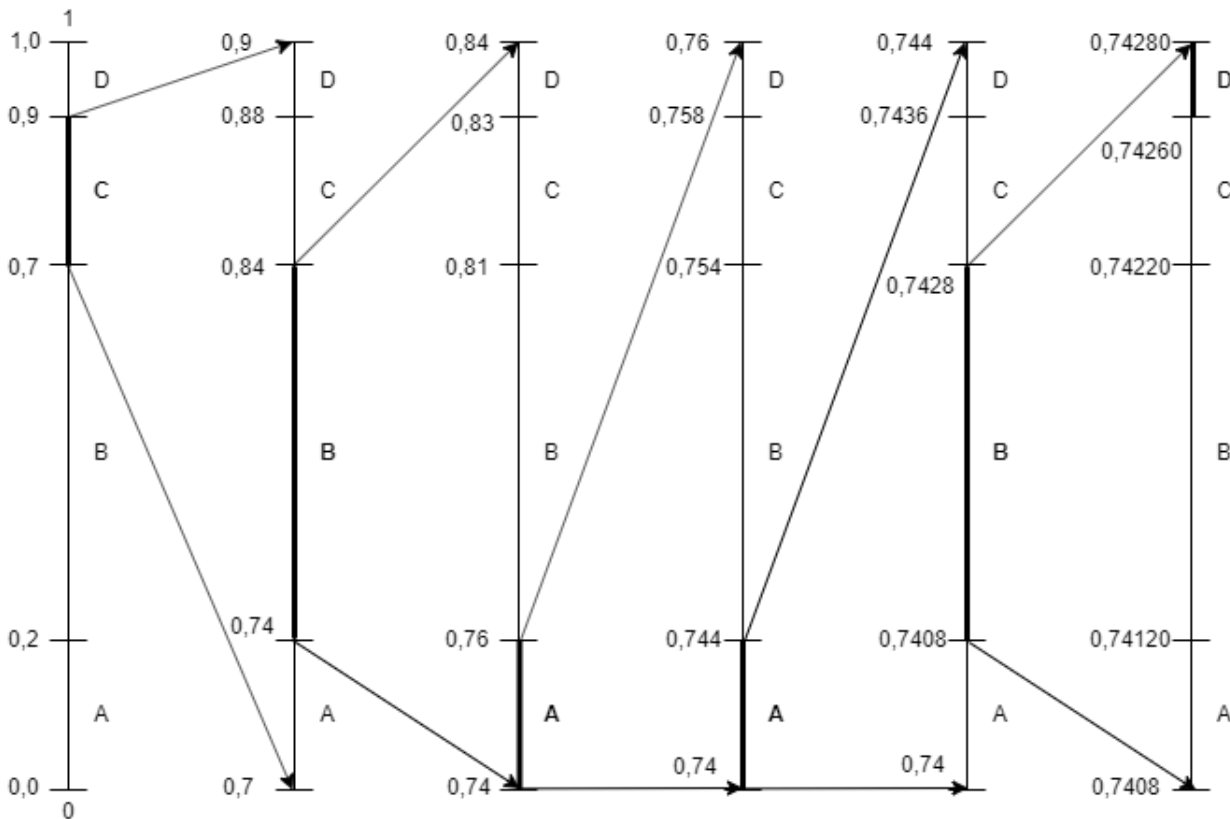
Iesūtīšanas veids: PDF uz epastu "kalvis.apsitis" domēnā "gmail.com".

Ieteicamā lasāmviela. [Blelloch2013], pp.16–19.

Sk. <http://linen-tracer-682.appspot.com/algorithms/references.html>

1. **Aritmētiskais kods.** Dota ziņojumu kopa $S = \{A, B, C, D\}$ ar attiecīgajām varbūtībām $\{0.2, 0.5, 0.2, 0.1\}$.

- (a) Parādīt, kā iegūt aritmētisko kodu 6 ziņojumu virknei CBAABD – uzkonstruēt tai atbilstošo intervālu $[l_6; l_6 + s_6] \in [0; 1]$ un atrast īsāko bitu virkni $d_1 d_2 \dots d_\ell$ (visi $d_k \in \{0, 1\}$, kur pierakstot binārā pieraksta daļskaitlim $D = 0.d_1 d_2 \dots d_\ell \dots$ galā jebkuru turpinājumu ar cipariem 0 vai 1, iegūtais skaitlis $d + \varepsilon$ pieder intervālam $[l_6; l_6 + s_6]$).



Aritmētisko kodu 6 ziņojumu virknei CBAABD var parādīt ar šo attēlu.

Sākotnēji bija dota ziņojumu kopa: $S = \{A, B, C, D\}$ un varbūtības: $p('A') = 0,2$; $p('B') = 0,5$; $p('C') = 0,2$; $p('D') = 0,1$.

Tika iegūts ziņojumu virknes CBAABD atbilstošais intervāls: $[0,74260; 0,74280)$.

$0,74260_{10} = 0,1011111000|0110110001$.

$0,74280_{10} = 0,1011111000|1010000001$.

Līdz ar to īsākā bitu virkne ir $D = 0,1011111000$, kas pieder intervālam: $[0,74260; 0,74280)$.

(b) Noteikt, kādu ziņojumu virkni alfabētā S kodē skaitlis $D'' = 0.0011101011_2$.

$D'' = 0,0011101011_2 = 0,229\,492\,187\,5_{10}$.

p	0.1	0.2	0.5	0.2
1	D	0.9	C	0.7
0.7	0.05	0.1	0.25	0.3
0.3	0.01	0.02	0.05	0.22
0.27	0.005	0.01	0.025	0.23
0.23	0.001	0.002	0.005	0.222
0.23	0.0001	0.0002	0.0005	0.222
0.2297	0.00005	0.0001	0.00025	0.2293
0.22955	0.00003	0.00005	0.00013	0.22935
0.229525	0.000005	0.00001	0.000025	0.229485
0.22951	0.000002	0.000005	0.000012	0.22949
0.229503	0.0000012	0.0000025	0.0000062	0.2294925
0.2294925	0.0000002	0.0000005	0.0000012	0.2294905

Šajā tabulā, kas ir pārcelta un transponēta no *Excel*, ir attēloti aprēķini, kas nepieciešami, lai atkodētu ziņojumu virkni: **BABADBBCBBAC...**

Ejot arvien dziļākos aprēķinos C turpina atkārtoties bezgalīgi.

2. Lempela-Ziva algoritms.

(a) Ar LZ78 metodi nokodēt tekstu “abracadabra, abracadabra”.

Dota virkne: “abracadabra, abracadabra”.

Sākumā vārdnīcā atrodas visi burti: **a, b, r, d, “,”, “_”**.

Komats un atstarpe arī ir burti, jo jākodē ir viss teksts, kas atrodas pēdējās.

Papildinām sākotnējo vārdnīcu, izmantojot LZ78 metodi.

Uzskatāmībai sadalām doto virkni pa fragmentiem, atdalot fragmentus ar vertikālām līnijām.

Virkne: |a|b|r|ac|ad|ab|ra|,| |abr|aca|da|br|a|

Fragmenti: |1|2|3|4 |5 |6 |7 |8|9|10 |11 |12|13|14|

Soļi: | | | |1 |2 |3 |4 | | |5 |6 |7 |8 | |9

Šajā tabulā ir izlaisti visi tie soļi, kad izrādās, ka virkne wk jau atrodas vārdnīcā.

Vārdnīca

Soļa nr.	w (garākais fragments, kas atrodams vārdnīcā)	k	Izvads	Pievieno vārdnīcai wk
1.	a	c	a	ac
2.	a	d	a	ad
3.	a	b	a	ab
4.	r	a	r	ra
5.	ab	r	ab	abr
6.	ac	a	ac	aca
7.	d	a	d	da
8.	b	r	b	br
9.			a	

Piezīme: k – burts, kas seko aiz jau zināmā vārda w.

Nokodētā virkne: a, b, r, 1, 2, 3, 4, “,”, “_”, 5, 6, 7, 8, a jeb citā pierakstā: abr1234,_5678a.

Tā kā literatūrā ir sastopami dažādi metodes LZ78 algoritmi sākumi, tad parādu savādāku sākumu teksta “abracadabra” kodēšanai. Tā kā šeit sākumā vārdnīcā nav neviens burts, tad “d” kā atsevišķs burts pēc šī algoritma neparādās. Tas radītu problēmas atkodējot burtu “d” kā prasīts MD1-2b.

Paraugs vārdnīcai:

In	Out	New phrases	
		Index	Phrase
		0	""
a	$(0, a)$	1	" a "
b	$(0, b)$	2	" b "
r	$(0, r)$	3	" r "
a			
c	$(1, c)$	4	" ac "
a			
d	$(1, d)$	5	" ad "
a			
b	$(1, b)$	6	" ab "
r			
a	$(3, a)$	7	" ra "
	$(0, EOF)$		

Avots: <http://ftp.project.ifi.uio.no/publications/cand-scient-theses/SHuseby/html/node43.html>

- (b) Atkodēt ar LZ78 metodi nokodētu tekstu $a, b, c, d, 2, 5, a, 6$, kur a , b un c apzīmē atbilstošos burtus, bet skaitļi – vārdnīcas virkņu numurus.

Atkodējamā virkne: $a, b, c, d, 2, 5, a, 6$.

Izmantojam MD1-2a vārdnīcu;

$a=a$

$b=b$

$c=c$

$d=d$

$2=ad$

$5=abr$

$a=a$

$6=aca$

Tātad atkodētā virkne sanāk: $abcdadabraaca$.

(c) Nokodēt (a) punkta tekstu “abracadabra, abracadabra” ar LZ77 metodi, kā logu lietojot visu nokodēto/atkodēto tekstu.

Dota virkne: “abracadabra, abracadabra”.

Jāizmanto metode LZ77, kā logs tiek izmantots: abcd25a6.

Virknes garums ir 22 simboli. Loga garums ir 8 simboli.

Pārvietojamies pa lentu. Sākumā galviņa atrodas lentas sākumā (pozīcijā 0).

↓abracadabra, abracadabra

1) $a \rightarrow (0, 0, a)$

a↓bracadabra, abracadabra

2) $b \rightarrow (0, 0, b)$

ab↓racadabra, abracadabra

3) $r \rightarrow (0, 0, r)$

abr↓acadabra, abracadabra

4) $a \rightarrow (3, 1, c)$ [a ir bijis. Sakrītošās virknes garums ir 1. Aiz a seko c. Pa kreisi no galviņas tuvākais a ir attālumā 3. Pēc tam pāriet pāri jaunajai virknei ac.]

abrac↓adabra, abracadabra

5) $a \rightarrow (2, 1, d)$

abracad↓abra, abracadabra

6) $a \rightarrow (7, 4, “,”)$

abracadabra↓, abracadabra

7) $“,” \rightarrow (0, 0, “__”)$

abracadabra, ↓abracadabra

8) $“__” \rightarrow (0, 0, a)$

Ja izmantojam loga garumu 14, tad turpinājums ir:

abracadabra, abracadabra↓

9) $a \rightarrow (13, 11, “”)$

Ja izmantojam loga garumu 8, tad turpinājums ir:

abrac|adabra, ↓abracadabra

9) $a \rightarrow (4, 6, c)$

abracadab|ra, abra↓cadabra

10) $c \rightarrow (0, 0, a)$

Sakārtojam rindas pēc alfabēta (pēc priekšpēdējā burtā ar skatu uz kreiso pusi).

usariirasula	a₁	l₁
lusariirasul	a₄	a₁
ulaalusariir	a₃	s₁
iirasulaalus	a₂	r₁
asulaalusari	i₂	r₂
rasulaalusari	i₁	i₂
sariirasulaa	l₁	u₁
alusariirasul	l₂	a₄
irasulaalusar	r₁	i₁
sulaalusariir	r₂	a₃
laalusariiras	s₁	u₂
riirasulaal	u₁	a₂
ariirasulaal	u₁	s₁
aalusariiras	u₂	l₂

Atbilde:

- 1) transformētā simbolu virkne ir: $l_1 a_1 s_1 r_1 r_2 i_2 u_1 a_4 i_1 a_3 u_2 a_2 s_1 l_2$
- 2) sākotnējās virknes pozīcija ir: 8.

(b) Kāds ir iepriekšējā piemērā iegūtās transformētās simbolu virknes pieraksts, izmantojot Move-to-Front kodēšanu?

Kā kodējamo virkni izmantojam MD1-3a rezultātu: lasrriuaiauasl

Kā kodēšanas metodi izmantojam Move-to-Front.

Alfabēts virknei $A = \{a, i, l, r, s, u\}$.

Sākam kodēšanu. Sanumurējam alfabētu.

0	1	2	3	4	5
a	i	l	r	s	u

Pirmais burts ir "l". Rakstam kodu 2. Pārceļam "l" uz priekšu un mainām tā kodu uz "0".

Pārējos burtus pārntimerējam.

l a s r r i u a i a u a s l

2

0	1	2	3	4	5
l	a	i	r	s	u

Nākošais burts ir “a”. Rakstām kodu 1. Pārceļam “a” uz priekšu un mainām tā kodu uz “0”.

Pārējos burtus pārunurējam.

l a s r r i u a i a u a s l

2 1

0	1	2	3	4	5
a	l	i	r	s	u

Turpinām kodēšanu.

l a s r r i u a i a u a s l

2 1 4

0	1	2	3	4	5
s	a	l	i	r	u

l a s r r i u a i a u a s l

2 1 4 4

0	1	2	3	4	5
r	s	a	l	i	u

Tā kā “r” atkārtojas, rakstām kodu 0. Pārkārtojumi nav jāveic.

l a s r r i u a i a u a s l

2 1 4 4 0

0	1	2	3	4	5
r	s	a	l	i	u

l a s r r i u a i a u a s l

2 1 4 4 0 4

0	1	2	3	4	5
i	r	s	a	l	u

l a s r r i u a i a u a s l

2 1 4 4 0 4 5

0	1	2	3	4	5
u	i	r	s	a	l

l a s r r i u a i a u a s l
2 1 4 4 0 4 5 4

0	1	2	3	4	5
a	u	i	r	s	l

l a s r r i u a i a u a s l
2 1 4 4 0 4 5 4 2

0	1	2	3	4	5
i	a	u	r	s	l

l a s r r i u a i a u a s l
2 1 4 4 0 4 5 4 2 1

0	1	2	3	4	5
a	i	u	r	s	l

l a s r r i u a i a u a s l
2 1 4 4 0 4 5 4 2 1 2

0	1	2	3	4	5
u	a	i	r	s	l

l a s r r i u a i a u a s l
2 1 4 4 0 4 5 4 2 1 2 1

0	1	2	3	4	5
a	u	i	r	s	l

l a s r r i u a i a u a s l
2 1 4 4 0 4 5 4 2 1 2 1 4

0	1	2	3	4	5
s	a	u	i	r	l

l a s r r i u a i a u a s l
2 1 4 4 0 4 5 4 2 1 2 1 4 5

0	1	2	3	4	5
l	s	a	u	i	r

Atbilde: Izmantojot Move-to-Front pierakstu virknes pieraksts ir: 21440454212145.

- (c) Pēc BW transformācijas pielietošanas tika iegūta simbolu virkne mmmrvvauuuibbbri. Kāda bija simbolu virkne pirms transformācijas (ņemot 4. virkni no atjaunotās tabulas)?

Dota virkne: mmmrvvauuuibbbri.

Virknes garums ir 17 simboli.

	F	L
0	a	m
1	b	m
2	b	m
3	b	r
4	i	v
5	i	v
6	i	a
7	m	u
8	m	u
9	m	u
10	r	i
11	r	i
12	u	b
13	u	b
14	u	b
15	v	r
16	v	i

Dekodēšanu veicam līdzīgi kā parādīts YouTube video:

<https://www.youtube.com/watch?v=iPCmvXJX2o0>

Strādājam ar F un L kolonām, kur L ir uzdevumā dotā virkne, kas pagriezta par kolonu un iet uz leju, bet kolona F ir tā pati virkne tikai sakārtota pēc alfabēta no augšas uz leju. Noliekot abas kolonas blakus tiek iegūti visi burtu pāri, kas sastopami virknē pirms transformācijas.

Uzdevumā dots, ka meklējamā ir 4. virkne, tādēļ dekodēšanu sākam ar ceturto rindu.

$F[4] = 'i'$. Atkodētā virkne ir “i”.

$L[10] = 'i'$ – atbilstošais i.

$F[10] = 'r'$. Atkodētā virkne ir “ir”.

$L[3] = 'r'$.

$F[3] = 'b'$. Atkodētā virkne ir “irb”.

$L[14] = 'b'$.

$F[14] = 'u'$. Atkodētā virkne ir “irbu”.

$L[9] = 'u'$.

$F[9] = 'm'$. Atkodētā virkne ir “irbum”.

$L[2] = 'm'$.

$F[2] = 'b'$. Atkodētā virkne ir “irbumb”.

$L[13] = 'b'$.

$F[13] = 'u'$. Atkodētā virkne ir “irbumbu”.

$L[8] = 'u'$.

$F[8] = 'm'$. Atkodētā virkne ir “irbumbum”.

$L[1] = 'm'$.

$F[1] = 'b'$. Atkodētā virkne ir “irbumbumb”.

$L[12] = 'b'$.

$F[12] = 'u'$. Atkodētā virkne ir “irbumbumbu”.

$L[7] = 'u'$.

$F[7] = 'm'$. Atkodētā virkne ir “irbumbumbum”.

$L[0] = 'm'$.

$F[0] = 'a'$. Atkodētā virkne ir “irbumbumbuma”.

$L[6] = 'a'$.

$F[6] = 'i'$. Atkodētā virkne ir “irbumbumbumai”.

$L[16] = 'i'$.

$F[16] = 'v'$. Atkodētā virkne ir “irbumbumbumaiv”.

$L[5] = 'v'$.

$F[5] = 'i'$. Atkodētā virkne ir “irbumbumbumaivi”.

$L[11] = 'i'$.

$F[11] = 'r'$. Atkodētā virkne ir “irbumbumbumaivir”.

$L[15] = 'r'$.

$F[15] = 'v'$. **Atkodētā virkne ir “irbumbumbumaivirv”.**

$L[4] = 'v'$.