

# Vidus eksāmens

Lietišķie algoritmi, 2020.g. rudens

Terminš: 2020-11-02

*Atrisinājumus lūdzam pārveidot par vienu PDF.*

---

## 1. uzdevums (Berouza-Vīlera transformācija)

(A) Izpildīt Berouza-Vīlera transformāciju ievades virknei jeb stringam **PANAMACANAL\$**, izmantojot ciklisko permutāciju parasto alfabētisko sakārtojumu (algoritmu sk. <https://bit.ly/37XTbd3>).

Veikt transformācijas rezultātam “move to front” kodējumu, pieņemot, ka alfabēts satur sekojošos burtus (uzskaitītajā secībā):

$$\mathcal{A} = \{\$, A, C, L, M, N, P\}$$

(B) Izpildīt Berouza-Vīlera transformāciju tam pašam stringam **PANAMACANAL\$**, izmantojot inversi leksikogrāfisko sakārtojumu (algoritmu sk. <https://bit.ly/3mIu7e9>, 37.lpp.).

(A) Izrakstām cikliskās permutācijas:

PANAMACANAL\$  
\$PANAMACANAL  
L\$PANAMACANA  
AL\$PANAMACAN  
NAL\$PANAMACA  
ANAL\$PANAMAC  
CANAL\$PANAMA  
ACANAL\$PANAM  
MACANAL\$PANA  
AMACANAL\$PAN  
NAMACANAL\$PA  
ANAMACANAL\$P

Sakārtojam alfabētiskā secībā:

\$PANAMACANAL  
ACANAL\$PANAM  
AL\$PANAMACAN  
AMACANAL\$PAN  
ANAL\$PANAMAC  
ANAMACANAL\$P  
CANAL\$PANAMA  
L\$PANAMACANA  
MACANAL\$PANA  
NAL\$PANAMACA  
NAMACANAL\$PA  
PANAMACANAL\$

Pēdējā kolonna ir transformācijas rezultāts: **LMNNCPAAAAA\$**.

Lietojam “Move to Front” kodējumu. Pašā sākumā alfabēts ir sākotnējā secībā. Iepriekšējā soļa ievadē sastaptais burts ikreiz pārceļo uz alfabēta sākumu.

Ievade	Alfabēts	Izvade
L	{\$, A, C, L, M, N, P}	3
M	{L, \$, A, C, M, N, P}	4
N	{M, L, \$, A, C, N, P}	5
N	{N, M, L, \$, A, C, P}	0
C	{N, M, L, \$, A, C, P}	5
P	{N, M, L, \$, A, C, P}	6
A	{P, N, M, L, \$, A, C}	5
A	{A, P, N, M, L, \$, C}	0
A	{A, P, N, M, L, \$, C}	0
A	{A, P, N, M, L, \$, C}	0
A	{A, P, N, M, L, \$, C}	0
\$	{A, P, N, M, L, \$, C}	5

Tātad pie dotā alfabēta  $\mathcal{A} = \{\$, A, C, L, M, N, P\}$  Berouza-Vīlera iekodēšanas rezultāts:

3, 4, 5, 0, 5, 6, 5, 0, 0, 0, 0, 5.

(B) Izrakstām cikliskās permutācijas:

PANAMACANAL \$  
 \$PANAMACANA L  
 L\$PANAMACAN A  
 AL\$PANAMACA N  
 NAL\$PANAMAC A  
 ANAL\$PANAMA C  
 CANAL\$PANAM A  
 ACANAL\$PANA M  
 MACANAL\$PAN A  
 AMACANAL\$PA N  
 NAMACANAL\$P A  
 ANAMACANAL\$ P

Sakārtojam inversi leksikogrāfiski virknītes (no sākuma līdz priekšpēdējai pozīcijai):

ANAMACANAL\$ P  
 AL\$PANAMACA N  
 ANAL\$PANAMA C  
 \$PANAMACANA L  
 ACANAL\$PANA M  
 AMACANAL\$PA N  
 NAL\$PANAMAC A  
 PANAMACANAL \$  
 CANAL\$PANAM A  
 L\$PANAMACAN A  
 MACANAL\$PAN A  
 NAMACANAL\$P A

Pēdējā kolonna ir transformācijas rezultāts: PNCLMNA\$AAAA.

□

## 2.uzdevums (Aritmētiskais kods).

Iekodēt **PANAMACANAL\$** ar aritmētisko kodu, ja visu septiņu alfabēta burtu varbūtības ir šādas:

\$	A	C	L	M	N	P
1/10	3/10	1/10	1/10	1/10	2/10	1/10

(A) Uzrakstīt pusatvērtu intervālu  $I_{12} = [c; d)$ , kurš izveidojas pēc visu 12 burtu iekodēšanas (var parādīt arī starprezultātus, lai varētu saņemt pozitīvu vērtējumu arī neuzmanības kļūdu gadījumā).

(B) Uzrakstīt bitu virknīti  $\beta = b_1b_2 \dots b_k$  ar  $k$  bitiem, ka skaitlis  $u$ , kura binārais pieraksts ir  $u = 0.b_1b_2 \dots b_k$  pieder iekodējamajam intervālam  $I_{12}$  un arī skaitlis  $v = u + \frac{1}{2^k}$  pieder  $I_{12}$ . Ja tādas virknītes ir vairākas, izvēlieties to, kura ir īsākā (kurai  $k$  vērtība ir minimālā), lai  $[u; v) \subseteq [c; d)$ .

Piemēram, bitu virknīte  $\beta = 010$  iekodē intervālu  $[u; v) = [1/4; 3/8)$ , bet virknīte  $\beta = 01$  iekodē intervālu  $[u; v) = [1/4; 1/2)$ . (Atkodēšanas algoritmam palīdz virknes/stringa beigu simbols \$, jo tiklīdz kā tas ir atkodēts un izvadīts, tad var beigt darbu.)

Izmantojot rekurentas sakarības, atrodam, ka

$$I_{12} = [0.935721517000; 0.935721517972).$$

□

## 3.uzdevums (Heminga kodi).

Atrast kļūdas (ja tādas ir) sekojošos Heminga koda ziņojumos:

(A) 1100000 (tas ir  $[7, 4, 1]$ -kods, bitu secība no  $x_{001}$  līdz  $x_{111}$ ).

(B) 0110011 (tas ir  $[7, 4, 1]$ -kods, bitu secība no  $x_{001}$  līdz  $x_{111}$ ).

(C) 001001111000101 (tas ir  $[15, 11, 1]$  Heminga kods, bitu secība – no  $x_{0001}$  līdz  $x_{1111}$ ).

## 4.uzdevums (Rīda-Solomona kods).

Divi kodētāji kodē to pašu bitu virkni ar Rīda-Solomona kodu:

(A) Pirmais kodētājs griež bitu virkni gabaliņos pa 4 bitiem, katru gabaliņu iekodē par skaitli  $a_i$  no 0 līdz 15, sakrāj  $n$  šādas vērtības un tad sūta  $n - 1$  pakāpes polinoma

$$p(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_1x + a_0$$

vērtības visos 17 punktos  $x \in \{0, 1, \dots, 16\}$ , veicot aprēķinus atbilstoši galīgā lauka  $GF(17)$  aritmētikai (tā ir aritmētika pēc moduļa 17).

(B) Otrais kodētājs griež bitu virkni gabaliņos pa 8 bitiem, katru iekodē par skaitli  $b_i$  no 0 līdz 255 un tad sūta  $n - 1$  pakāpes polinoma

$$q(x) = b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \dots + b_1x + b_0$$

vērtības visos 257 punktos  $x \in \{0, 1, \dots, 256\}$ , veicot aprēķinus atbilstoši galīgā lauka  $GF(257)$  aritmētikai (tā ir aritmētika pēc pirmkaitļa 257 moduļa).

Abos gadījumos atrast, cik kļūdas var izlabot (atkarībā no parametra  $n$ ). Par kļūdu uzskatām situāciju, kad attiecīgo polinoma  $p(x)$  vai  $q(x)$  vērtību saņemējam neizdevās saņemt (saņemējam ir zināms, kuras vērtības pa ceļam sabojājās, bet viņam nav zināms, kādas bija šīs vērtības).