2. mājasdarbs

Lietišķie algoritmi, 2020.g. rudens
Termiņš: 2020-11-09

Atrisinājumus lūdzam pārveidot par vienu PDF.

Vairāki uzdevumi šajā mājasdarbā iespaidojušies no MIT Open Courseware: `https://bit.ly/3dabHyG` and `https://bit.ly/36xvx4e`.

1.uzdevums (Kodēšana ar tabulu). Alise izveidoja $[5, 2, 1]$-kodu 2-bitu datiem ($k = 2$), ko pārraida ar 5-bitu kodējumiem ($n = 5$), kas atļauj 1 kļūdas izlabošanu, jo Heminga attālums starp jebkuriem diviem kodējumiem ir vismaz 3. Pirmie divi biti katrā kodējumā pārraida divus *derīgo datu* (payload) bitus; vēl trīs biti pievienoti kļūdu aizsardzībai. Diemžēl, Alises suns sagrauza viņas piezīmes un iznīcināja daļu no kodējumu tabulas (parādīts ar jautājuma zīmēm). Jūsu uzdevums ir atjaunot kodu, kas apmierina augšminētās prasības.

Ievade		Kodējums				
`00`	$\to$	0	0	?	?	?
`01`	$\to$	0	1	?	?	?
`10`	$\to$	1	0	?	?	?
`11`	$\to$	1	1	0	0	?

Table 1: Alises kodu tabula.

(a) Atrast kaut vienu veidu, kā atjaunot kodu tabulu, kas parādītu vienu veidu, kā nokodēt katru no 4 divu bitu virknītēm, kas var būt ievadē. (Vai arī pamatojiet, ka no šīs tabulas $[5, 2, 1]$-kodu nevar iegūt.)

(b) No 32 iespējamām 5-bitu virknītēm, cik daudzām ir Heminga attālums tieši 1 līdz kādam no iekodējumiem; t.i. tās var pārlabot uz tuvāko atļauto vērtību, pieņemot, ka tās radīja viena kļūda?

(c) Cik daudzas no 5-bitu virknītēm varēja rasties tikai tad, ja ir vairāk nekā 1 kļūda?

(d) Jūsu priekšnieks uzskata, ka elektrības taupīšanas apstākļos sūtīt 1-bitu ir dārgāk nekā 0-bitu. Vai iespējams samazināt 1-bitu skaitu, kas ir Jūsu iekodējumu tabulā? (Tikai jautājumzīmes drīkst mainīt savas bitu vērtības; visiem pārējiem bitiem, kas nav sagrauzti, jāpaliek tādiem, kādi tie tabulā ir.)

2.uzdevums (Lineārie algoritmi) Apzīmējam matricu H:

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

(a) Atrast matricu G ar lielākajiem izmēriem (un ar lineāri neatkarīgām kolonnām), kurai $G \cdot H$ ir matrica, kura satur tikai nulles, ja visas matricu darbības veic pēc 2 moduļa.

(b) Ko var apgalvot par G ģenerēto kodu: Kādi ir mazākie iespējamie Heminga attālumi starp $\mathbf{x}_1^T \cdot G$ un $\mathbf{x}_2^T \cdot G$, kur $\mathbf{x}_1^T, \mathbf{x}_2^T \in \{0,1\}^4$ ir jebkuras 4-bitu virknes kas pierakstītas kā rindas vektori.

Piezīme. Matricas G kolonnas saucam par *lineāri neatkarīgām* ja katrai netukšai G kolonnu apakškopai, šo kolonnu summa nevar būt vektors, kas satur tikai nulles (saskaitot pēc 2 moduļa). Sk. teoriju `https://bit.ly/37Fv9mJ`.

3.uzdevums (Diskrētais kosinusu pārveidojums).
Dota funkcija $f(x)$, kas definēta argumentiem $x = 0, 1, \ldots, N-1$. Par 1-dimensionālu DCT (diskrēto kosinusu pārveidojumu jeb *discrete cosine transform*) sauksim funkciju $F(u)$, kas definēta tām pašām argumenta vērtībām $u = 0, 1, \ldots, N-1$ ar šādām vienādībām:

$$F(u) = \sqrt{\frac{2}{N}} \cdot \lambda_u \cdot \sum_{x=0}^{N-1} \cos\left(\frac{\pi u}{N}\left(x + \frac{1}{2}\right)\right) \cdot f(x),$$

kur $u = 0, 1, \ldots, N-1$ un $\lambda_u = \frac{1}{\sqrt{2}}$ (pie $u = 0$) un $\lambda_u = 1$ (pie $u > 0$).
Sk. `https://bit.ly/3mj8h0A`.
Par inverso diskrēto kosinusu pārveidojumu sauksim atgriešanos no funkcijas $F(u)$ atpakaļ pie funkcijas $f(x)$, ko definē ar šādām vienādībām:

$$f(x) = \sqrt{\frac{2}{N}} \sum_{u=0}^{N-1} \lambda_u \cdot \cos\left(\frac{\pi u}{N}\left(x + \frac{1}{2}\right)\right) \cdot F(u),$$

kur $x = 0, 1, \ldots, N-1$ un λ_u definēti tāpat kā agrāk.

(a) Atrast diskrēto kosinusu pārveidojumu $F(u)$ punktā $u = 3$ funkcijai $f(x) = (N-1) - x$, kur $N = 8$. Atbildi noapaļot līdz 4 cipariem aiz komata.

(b) Aprēķināt inverso diskrēto kosinusu pārveidojumu $f(x)$ visiem punktiem $x = 0, 1, 2, 3, 4, 5, 6, 7$ no funkcijas $F(u)$, kas uzdota ar sekojošām $N = 8$ vērtībām:

$$(F(0), F(1), F(2), F(3), F(4), F(5), F(6), F(7)) =$$
$$= (57.9828, -6.4423, 0, -0.6735, 0, -0.2009, 0, -0.0507).$$

Atbildi noapaļot līdz 4 cipariem aiz komata.

4.uzdevums (Rīda-Solomona kods).

1. Izmantojot galīgu lauku ar 7 elementiem GF(7) kodējam ziņojumus no 7 simbolu alfabēta $\{0, 1, 2, \ldots, 6\}$ ar 3.pakāpes polinomiem $f(n)$, pārraidot 7 polinoma vērtības $(f(0), \ldots, f(6))$ visos galīgā lauka GF(7) punktos. Dažas no šīm septiņām vērtībām var pa ceļam sabojāties (tikt aizstātas ar citu GF(7) elementu). Kāds ir maksimālais kļūdu skaits, pie kura iespējams viennozīmīgi atjaunot sākotnējo ziņojumu? Pamatot, kāpēc ir iespējams koriģēt šādu kļūdu skaitu un kāpēc nav iespējams koriģēt lielāku kļūdu skaitu.

2. Galīga lauka GF(2^3) elementus

$$\{0, 1, t, t+1, t^2, t^2+1, t^2+t, t^2+t+1\}$$

apzīmējam attiecīgi ar bitu virknēm

$$\{\texttt{000}, \texttt{001}, \texttt{010}, \texttt{011}, \texttt{100}, \texttt{101}, \texttt{110}, \texttt{111}\}.$$

3-pakāpes polinoms

$$p(x) = \texttt{101} \cdot x^3 + \texttt{100} \cdot x^2 + \texttt{001} \cdot x + \texttt{010}$$

domāts, lai pārraidītu ziņojumu virknīti
`101.100.001.010`. Atrast polinoma vērtību $p(\texttt{011})$, kur $\texttt{011} \in \mathrm{GF}(2^3)$.

Piezīme: Faktiski Rīda-Solomona kļūdu labošanas kodā vajadzētu sūtīt visas 8 polinoma vērtības, bet šajā vingrinājumā pietiek izrēķināt $p(x)$ tikai pie $x = \texttt{011}$.

5.uzdevums (I-Iespēja)

Jūs veidojat produktu, kas izmanto *Taisnstūrveida kodu* (rectangular code) (sk. 48.lpp. no `https://bit.ly/2M5ptGR`) lai nodrošinātu kritisko bitu pareizību, kurus sūta pa trokšņainu kanālu. Jūsu risinājumā katru bloku ar 9 "derīgo datu" (payload) bitiem aizsargā ar Taisnstūrveida kodu. Jūsu metode katrus deviņus derīgo datu bitus (D0, ..., D8) aizsargā ar septiņiem kļūdu labošanas bitiem (PR0, PR1, PR2, PC0, PC1, PC2, P0). Šie biti derīgo datu bitu summas (pēc 2 moduļa) pa rindiņām vai kolonnām (bet $P0$ ir visu datu bitu summa).
Kļūdu labošanas metodei jāizpilda 2 prasības:

- Atrast un izlabot kļūdu jebkurā vienā bitā no deviņiem (D0, ..., D8).
- Atpazīt/detektēt kļūdas jebkuros divos bitos no deviņiem (iespējams, nepasakot, kuri ir kļūdainie biti).

D0	D1	D2	PR0
D3	D4	D5	PR1
D6	D7	D8	PR2
PC0	PC1	PC2	P0

Table 2: Taisnstūrveida kods.

Bens – viens no Jūsu kolēģiem – pārbaudījis Jūsu metodi, ierosina to mainīt tā, ka Jūs nepārraidāt paritātes bitus PR0 un PC0, tikai tos četrus bitus, kas saistīti ar citām rindām un kolonnām (PR1, PR2 un PC1, PC2), kā arī kopējo paritātes bitu ($P0$). Viņš apgalvo, ka arī šāds kods izpildīs abas prasības un efektīvāk izmantos sakaru kanālu.

(a) Vai visas viena bita kļūdas var atrast un izlabot (ar Bena piedāvāto izmaiņu).
Pamatojiet, ka vienmēr var vai atrodiet pretpiemēru.

(b) Vai visas divu bitu kļūdas var atrast un izlabot (ar Bena piedāvāto izmaiņu).
Pamatojiet, ka vienmēr var vai atrodiet pretpiemēru.

(c) Vai visas divu bitu kļūdas var atpazīt kā divu bitu kļūdas – nenosakot, kuros bitos bija kļūda (ar Bena piedāvāto izmaiņu).
Pamatojiet, ka vienmēr var vai atrodiet pretpiemēru.