Lietišķie algoritmi – 2. mājas darbs

Termiņš: 2019. gada 28. oktobrī 23:59:59 pēc Austrumeiropas ziemas laika (EET) jeb UTC+2.
Iesūtīšanas veids: PDF uz epastu "kalvis.apsitis" domēnā "gmail.com".

1. **Diskrētā krāsu plakne YCbCr.** Koordinātes (Y, Cb, Cr) aprēķina no koordinātēm (R, G, B) atbilstoši sekojošai vektoru algebras sakarībai:

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 65.48 & 128.55 & 24.97 \\ -37.78 & -74.16 & 111.93 \\ 111.96 & -93.75 & -18.21 \end{pmatrix} \begin{pmatrix} \frac{1}{255} & 0 & 0 \\ 0 & \frac{1}{255} & 0 \\ 0 & 0 & \frac{1}{255} \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 16 \\ 128 \\ 128 \end{pmatrix}.$$

Sk. `https://onlinelibrary.wiley.com/doi/full/10.1002/col.22291`.
Atrast YCbCr koordinātes zemāk minētajām krāsām, kas uzdotas (R, G, B) krāsu plaknē (un noapaļot visas (Y, Cb, Cr) koordinātes līdz tuvākajam veselajam skaitlim).

(a) Baltai krāsai `#FFFFFF` jeb $(R, G, B) = (255, 255, 255)$.
(b) Ciāna krāsai `#00FFFF` jeb $(R, G, B) = (0, 255, 255)$.
(c) Magentas krāsai `#FF00FF` jeb $(R, G, B) = (255, 0, 255)$.
(d) Dzeltenai krāsai `#FFFF00` jeb $(R, G, B) = (255, 255, 0)$.
(e) Melnai krāsai `#000000` jeb $(R, G, B) = (0, 0, 0)$.

2. **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:

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

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://en.wikipedia.org/wiki/Discrete_cosine_transform#DCT-II`.

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) Aprēķināt 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.
(Turpinājums lapas otrā pusē.)

3. **Heminga kodi.** Uzrakstām skaitļa $\pi = 3.14159\ldots$ pierakstu divnieku skaitīšanas sistēmā un grupējam tā ciparus aiz komata - divas grupas pa 7 un divas grupas pa 15:

$$\pi = 11.\underbrace{0010010}\underbrace{0001111}\underbrace{110110101010001}\underbrace{000100001011010}\ldots_2.$$

Grupējam šī skaitļa ciparus aiz komata grupās pa 7 (un pēc tam arī pa 15), lai iegūtu ziņojumus. Atrast kļūdas (ja tādas ir) atbilstošajos Heminga koda ziņojumos:

(a) Heminga koda ziņojums `0010010` (7-bitu Heminga kods, bitu secība apgriezti leksikogrāfiska – no x_{111} līdz x_{001}) – atrast kļūdaino pozīciju, ja tāda ir, un uzrakstīt šo 7-bitu kodu bez kļūdām.

(b) Heminga koda ziņojums `0001111` (7-bitu Heminga kods) – izlabot kļūdas, ja tās ir, un uzrakstīt 4 ziņojuma bitus $x_1x_2x_3x_4$.

(c) Heminga koda ziņojums `110110101010001` (15-bitu Heminga kods, bitu secība apgriezti leksikogrāfiska – no x_{1111} līdz x_{0001}) – atrast kļūdaino pozīciju, ja tāda ir, un uzrakstīt šo 15-bitu kodu bez kļūdām.

(d) Heminga koda ziņojums `000100001011010` (15-bitu Heminga kods) – izlabot kļūdas, ja tās ir, un uzrakstīt visus ziņojuma bitus (ziņojumu bitu secība apgriezti leksikogrāfiska $x_{1111}\ldots x_{0011}$).

4. **Rīda-Solomona kods.**

(a) Izmantojot galīgu lauku GF(7) kodējam ziņojumus no 7 simbolu alfabēta $\{0, 1, 2, \ldots, 6\}$ ar 3 pakāpes polinomiem, pārraidot 7 polinoma vērtības $(f(0), \ldots, f(7))$ visos galīgā lauka GF(7) punktos. 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.

(b) Galīga lauka $\mathrm{GF}(2^3)$ elementus $\left\{0, 1, t, t+1, t^2, t^2+1, t^2+t, t^2+t+1\right\}$ apzīmējam attiecīgi ar bitu virknēm $\{\mathtt{000}, \mathtt{001}, \mathtt{010}, \mathtt{011}, \mathtt{100}, \mathtt{101}, \mathtt{110}, \mathtt{111}\}$.
3-pakāpes polinoms $p(x) = \mathtt{101}\cdot x^3 + \mathtt{100}\cdot x^2 + \mathtt{001}\cdot x + \mathtt{010}$ ir domāts, lai pārraidītu ziņojumu virknīti `101.100.001.010`. Atrast polinoma vērtību $p(\mathtt{011})$, kur $\mathtt{011} \in \mathrm{GF}(2^3)$.
Piezīme: Faktiski Rīda-Solomona kļūdu labošanas kodā vajadzētu sūtīt visas 8 polinoma vērtības, bet šajā vingrinājumā pietiek izrēķināt $p(x)$ tikai pie $x = \mathtt{011}$.

5. **I-iespēja (atzīmei 10).**

(a) Uzrakstīt grafu kodu ar 9 ziņojuma bitiem $x_1, \ldots, x_9$ un pēc iespējas mazāku skaitu kontrolbitu $y_1, \ldots, y_k$ tā, lai kods spētu atjaunot jebkurus 2 pazaudētus ziņojuma bitus x_i.

(b) Pamatot, ka mazāks kontrolbitu skaits nav iespējams.

(c) Uzrakstīt grafu kodu ar 9 ziņojuma bitiem $x_1, \ldots, x_9$ un pēc iespējas mazāku skaitu kontrolbitu $y_1, \ldots, y_k$ tā, lai kods spētu atjaunot jebkurus 3 pazaudētus ziņojuma bitus x_i.