Lietišķie algoritmi – Semestra vidus eksāmena atrisinājumi (2019-10-29)

1.uzdevums (1+1+1+2 punkti): Spēlētājs X vienā gājienā izņem no urnas trīs kartiņas. Pieņemsim, ka urna ir ļoti liela, kartiņas tajā nekad nebeidzas un ir vienādas varbūtības izķeksēt jebkuru no burtiem `A`, `B` vai `C`; citu burtu urnā nav. Pēc tam X sakārto trīs kartiņas alfabētiskā secībā un nosūta spēlētājam Y ziņojumu – to burtu, kurš pēc sakārtošanas bija pirmais. (Piemēram, ja izķeksētie burti ir `"CBC"`, tad pēc sakārtošanas tie būs `"BCC"` un X nosūta ziņojumu `"B"`.)

(a) Kāds ir informācijas saturs ziņojumam `A`?

(b) Kāds ir informācijas saturs ziņojumam `B`?

(c) Kāds ir informācijas saturs ziņojumam `C`?

(d) Kāda ir entropija jebkuram vienam ziņojumam, ko X nosūta Y saskaņā ar augšminēto procedūru?

Atrisinājums:

(a) No visiem 27 variantiem, kā izvilkt kartiņas, ir 9 tādi, kas sākas ar burtu `A`. Starp atlikušajiem 18 trešajai daļai (jeb 6 variantiem) `A` ir otrajā pozīcijā. Visbeidzot, starp atlikušajiem $27-9-6=12$ ir trešdaļa jeb 4 tādi, kam `A` ir trešajā pozīcijā. Tātad ziņojuma `A` varbūtība ir $(9+6+4)/(27) = 19/27$.
Informācijas saturs: $-\log_2(P(\texttt{A})) \approx 0.507$.

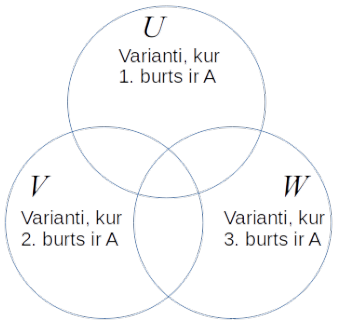
(b) Ziņojuma `B` varbūtība ir 7/27 (no 1 atņem ziņojumu `A` un `C` varbūtības).
Informācijas saturs ir $-\log_2(P(\texttt{B})) \approx 1.948$.

(c) Ziņojums `C` var rasties vienīgi izvelkot kartiņas `CCC`, tā varbūtība ir $p(\texttt{C}) = 1/27$.
Informācijas saturs ir $-\log_2(P(\texttt{C})) \approx 4.754$.

(d) Entropija ir

$$-P(\texttt{A})\log_2(P(\texttt{A})) - P(\texttt{B})\log_2(P(\texttt{B})) - P(\texttt{C})\log_2(P(\texttt{C})) =$$
$$= (19/27)\cdot 0.507 + (7/27)\cdot 1.948 + (1/27)\cdot 4.755 \approx 1.038.$$

Piezīme. Variantu skaitu, kuros ir vismaz viens burts `A` var saskaitīt arī citādi. Ar U apzīmējam visus tos variantus, kuros pirmais burts ir `"A"`, ar V – tos, kuros otrais burts ir `"A"`, un ar W – tos, kuros trešais burts ir `"A"`. Viegli redzēt, ka elementu skaits kopās $|U| = |V| = |W| = 9$, elementu skaits kopu šķēlumos ir $|U \cap V| = |U \cap W| = |V \cap W| = 3$ (ir 3 tādi varianti, kuros pirmais **un** otrais burts ir `"A"`). Savukārt visu trīs kopu šķēlums $|U \cap V \cap W| = 1$ (atbilst gadījumam, kad visi trīs burti ir `"A"`).



Sk. `https://en.wikipedia.org/wiki/Inclusion%E2%80%93exclusion_principle` (ieslēgšanas-izslēgšanas principu), kas ļauj noteikt elementu skaitu visu šo trīs kopu apvienojumā:

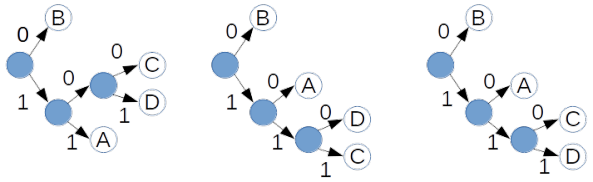
$$|U \cup V \cup W| = |U| + |V| + |W| - |U \cap V| - |U \cap W| - |V \cap W| + |U \cap V \cap W|\,.$$

$$|U \cup V \cup W| = 9 + 9 + 9 - 3 - 3 - 3 + 1 = 19.$$

2.uzdevums (3 punkti - jebkāds pareizs Hafmana koks, 3 punkti - koks kanoniskajā formā): Hafmana koku sauksim par *kanonisku*, ja izpildās sekojošas īpašības:

- Visi kanoniskā koka zari (ceļi no saknes līdz lapām/ziņojumiem) veido garumus, kuri ir nedilstošā secībā, skaitot no augšas uz leju.
- Vienāda garuma zariem ziņojumi izkārtoti ziņojumu alfabētiskā secībā.

Attēla 1.piemērā zaru garumi nav nedilstošā secībā (zars `11` uz lapu `A` ir garumā 2, virs tā divi zari garumā 3). 2.piemērā `C`, `D` ir ar vienādi gariem kodavārdiem, bet nav alfabētiskā secībā.
Vienīgi 3.piemērā Hafmana koks ir kanonisks.



5 ziņojumu kopai $\{\texttt{A}, \texttt{B}, \texttt{C}, \texttt{D}, \texttt{E}\}$, kuru varbūtības ir attiecīgi $\left\{\frac{1}{15}, \frac{2}{15}, \frac{3}{15}, \frac{4}{15}, \frac{5}{15}\right\}$, atrast un uzzīmēt kanonisku Hafmana koku.

Atrisinājums:

3.uzdevums (4 punkti): Izmantojot LZ78 algoritmu 6 simbolu alfabētam $\{\texttt{C}, \texttt{E}, \texttt{L}, \texttt{N}, \texttt{S}, \texttt{U}\}$, izveidot tabulu un nokodēt sekojošu 15 simbolu ziņojumu: `SUCCESSLESSNESS`. Tabulā attēlot soļa numuru, w - garāko vārdnīcā jau atrodamo simbolu virkni, k - virknei w sekojošo simbolu, algoritma izvadi un vārdnīcai attiecīgajā solī pievienojamo vārdu.

Solis	w	k	Izvade	Pievieno vārdnīcai
...	...	...	...	...

Atrisinājums:

4.uzdevums (6 punkti): Veikt inverso Berouza-Vīlera transformāciju 17 simbolu virknītei: `SETTNRSDDIESIEESN`, ja zināms, ka kodētais vārds ir pats pirmais starp leksikogrāfiski sakārtotajām cikliskajām permutācijām.

Atrisinājums: Berouza-Vīlera atkodēšanu veicam vairākos soļos: Vienā solī pierakstām priekšā matricai kolonnu, kas sakrīt ar atkodējamo virkni (solis `add`), pēc tam sakārtojam rindiņas alfabētiski (solis `sort`). Minēto soli atkārto tik daudz reižu, cik simbolu ir jāatkodē.

$$\xrightarrow{\text{add}} \begin{pmatrix} \texttt{S}\\ \texttt{E}\\ \texttt{T}\\ \texttt{T}\\ \texttt{N}\\ \texttt{R}\\ \texttt{S}\\ \texttt{D}\\ \texttt{D}\\ \texttt{I}\\ \texttt{E}\\ \texttt{S}\\ \texttt{I}\\ \texttt{E}\\ \texttt{E}\\ \texttt{S}\\ \texttt{N} \end{pmatrix} \xrightarrow{\text{sort}} \begin{pmatrix} \texttt{D}\\ \texttt{D}\\ \texttt{E}\\ \texttt{E}\\ \texttt{E}\\ \texttt{E}\\ \texttt{I}\\ \texttt{I}\\ \texttt{N}\\ \texttt{N}\\ \texttt{R}\\ \texttt{S}\\ \texttt{S}\\ \texttt{S}\\ \texttt{S}\\ \texttt{T}\\ \texttt{T} \end{pmatrix} \xrightarrow{\text{add}} \begin{pmatrix} \texttt{SD}\\ \texttt{ED}\\ \texttt{TE}\\ \texttt{TE}\\ \texttt{NE}\\ \texttt{RE}\\ \texttt{SI}\\ \texttt{DI}\\ \texttt{DN}\\ \texttt{IN}\\ \texttt{ER}\\ \texttt{SS}\\ \texttt{IS}\\ \texttt{ES}\\ \texttt{ES}\\ \texttt{ST}\\ \texttt{NT} \end{pmatrix} \xrightarrow{\text{sort}} \begin{pmatrix} \texttt{DI}\\ \texttt{DN}\\ \texttt{ED}\\ \texttt{ER}\\ \texttt{ES}\\ \texttt{ES}\\ \texttt{IN}\\ \texttt{IS}\\ \texttt{NE}\\ \texttt{NT}\\ \texttt{RE}\\ \texttt{SD}\\ \texttt{SI}\\ \texttt{SS}\\ \texttt{ST}\\ \texttt{TE}\\ \texttt{TE} \end{pmatrix} \xrightarrow{\text{add}} \begin{pmatrix} \texttt{SDI}\\ \texttt{EDN}\\ \texttt{TED}\\ \texttt{TER}\\ \texttt{NES}\\ \texttt{RES}\\ \texttt{SIN}\\ \texttt{DIS}\\ \texttt{DNE}\\ \texttt{INT}\\ \texttt{ERE}\\ \texttt{SSD}\\ \texttt{ISI}\\ \texttt{ESS}\\ \texttt{EST}\\ \texttt{STE}\\ \texttt{NTE} \end{pmatrix} \xrightarrow{\text{sort}} \begin{pmatrix} \texttt{DIS}\\ \texttt{DNE}\\ \texttt{EDN}\\ \texttt{ERE}\\ \texttt{ESS}\\ \texttt{EST}\\ \texttt{INT}\\ \texttt{ISI}\\ \texttt{NES}\\ \texttt{NTE}\\ \texttt{RES}\\ \texttt{SDI}\\ \texttt{SIN}\\ \texttt{SSD}\\ \texttt{STE}\\ \texttt{TED}\\ \texttt{TER} \end{pmatrix} \xrightarrow{\text{add}} \begin{pmatrix} \texttt{SDIS}\\ \texttt{EDNE}\\ \texttt{TEDN}\\ \texttt{TERE}\\ \texttt{NESS}\\ \texttt{REST}\\ \texttt{SINT}\\ \texttt{DISI}\\ \texttt{DNES}\\ \texttt{INTE}\\ \texttt{ERES}\\ \texttt{SSDI}\\ \texttt{ISIN}\\ \texttt{ESSD}\\ \texttt{ESTE}\\ \texttt{STED}\\ \texttt{NTER} \end{pmatrix} \xrightarrow{\text{sort}} \begin{pmatrix} \texttt{DISI}\\ \texttt{DNES}\\ \texttt{EDNE}\\ \texttt{ERES}\\ \texttt{ESSD}\\ \texttt{ESTE}\\ \texttt{INTE}\\ \texttt{ISIN}\\ \texttt{NESS}\\ \texttt{NTER}\\ \texttt{REST}\\ \texttt{SDIS}\\ \texttt{SINT}\\ \texttt{SSDI}\\ \texttt{STED}\\ \texttt{TEDN}\\ \texttt{TERE} \end{pmatrix}$$

$$\xrightarrow{\text{add}}\begin{pmatrix}\texttt{SDISI}\\ \texttt{EDNES}\\ \texttt{TEDNE}\\ \texttt{TERES}\\ \texttt{NESSD}\\ \texttt{RESTE}\\ \texttt{SINTE}\\ \texttt{DISIN}\\ \texttt{DNESS}\\ \texttt{INTER}\\ \texttt{EREST}\\ \texttt{SSDIS}\\ \texttt{ISINT}\\ \texttt{ESSDI}\\ \texttt{ESTED}\\ \texttt{STEDN}\\ \texttt{NTERE}\end{pmatrix}\xrightarrow{\text{sort}}\begin{pmatrix}\texttt{DISIN}\\ \texttt{DNESS}\\ \texttt{EDNES}\\ \texttt{EREST}\\ \texttt{ESSDI}\\ \texttt{ESTED}\\ \texttt{INTER}\\ \texttt{ISINT}\\ \texttt{NESSD}\\ \texttt{NTERE}\\ \texttt{RESTE}\\ \texttt{SDISI}\\ \texttt{SINTE}\\ \texttt{SSDIS}\\ \texttt{STEDN}\\ \texttt{TEDNE}\\ \texttt{TERES}\end{pmatrix}\xrightarrow{\text{add}}\begin{pmatrix}\texttt{SDISIN}\\ \texttt{EDNESS}\\ \texttt{TEDNES}\\ \texttt{TEREST}\\ \texttt{NESSDI}\\ \texttt{RESTED}\\ \texttt{SINTER}\\ \texttt{DISINT}\\ \texttt{DNESSD}\\ \texttt{INTERE}\\ \texttt{ERESTE}\\ \texttt{SSDISI}\\ \texttt{ISINTE}\\ \texttt{ESSDIS}\\ \texttt{ESTEDN}\\ \texttt{STEDNE}\\ \texttt{NTERES}\end{pmatrix}\xrightarrow{\text{sort}}\begin{pmatrix}\texttt{DISINT}\\ \texttt{DNESSD}\\ \texttt{EDNESS}\\ \texttt{ERESTE}\\ \texttt{ESSDIS}\\ \texttt{ESTEDN}\\ \texttt{INTERE}\\ \texttt{NESSDI}\\ \texttt{NTERES}\\ \texttt{ISINTE}\\ \texttt{RESTED}\\ \texttt{SDISIN}\\ \texttt{SINTER}\\ \texttt{SSDISI}\\ \texttt{STEDNE}\\ \texttt{TEDNES}\\ \texttt{TEREST}\end{pmatrix}\xrightarrow{\text{add}}\begin{pmatrix}\texttt{SDISINT}\\ \texttt{EDNESSD}\\ \texttt{TEDNESS}\\ \texttt{TERESTE}\\ \texttt{NESSDIS}\\ \texttt{RESTEDN}\\ \texttt{SINTERE}\\ \texttt{DISINTE}\\ \texttt{DNESSDI}\\ \texttt{INTERES}\\ \texttt{ERESTED}\\ \texttt{SSDISIN}\\ \texttt{ISINTER}\\ \texttt{ESSDISI}\\ \texttt{ESTEDNE}\\ \texttt{STEDNES}\\ \texttt{NTEREST}\end{pmatrix}\xrightarrow{\text{sort}}\begin{pmatrix}\texttt{DISINTE}\\ \texttt{DNESSDI}\\ \texttt{EDNESSD}\\ \texttt{ERESTED}\\ \texttt{ESSDISI}\\ \texttt{ESTEDNE}\\ \texttt{INTERES}\\ \texttt{ISINTER}\\ \texttt{NESSDIS}\\ \texttt{NTEREST}\\ \texttt{RESTEDN}\\ \texttt{SDISINT}\\ \texttt{SINTERE}\\ \texttt{SSDISIN}\\ \texttt{STEDNES}\\ \texttt{TEDNESS}\\ \texttt{TERESTE}\end{pmatrix}$$

$$\xrightarrow{\text{add}}\begin{pmatrix}\texttt{SDISINTE}\\ \texttt{EDNESSDI}\\ \texttt{TEDNESSD}\\ \texttt{TERESTED}\\ \texttt{NESSDISI}\\ \texttt{RESTEDNE}\\ \texttt{SINTERES}\\ \texttt{DISINTER}\\ \texttt{DNESSDIS}\\ \texttt{INTEREST}\\ \texttt{ERESTEDN}\\ \texttt{SSDISINT}\\ \texttt{ISINTERE}\\ \texttt{ESSDISIN}\\ \texttt{ESTEDNES}\\ \texttt{STEDNESS}\\ \texttt{NTERESTE}\end{pmatrix}\xrightarrow{\text{sort}}\begin{pmatrix}\texttt{DISINTER}\\ \texttt{DNESSDIS}\\ \texttt{EDNESSDI}\\ \texttt{ERESTEDN}\\ \texttt{ESSDISIN}\\ \texttt{ESTEDNES}\\ \texttt{INTEREST}\\ \texttt{ISINTERE}\\ \texttt{NESSDISI}\\ \texttt{NTERESTE}\\ \texttt{RESTEDNE}\\ \texttt{SDISINTE}\\ \texttt{SINTERES}\\ \texttt{SSDISINT}\\ \texttt{STEDNESS}\\ \texttt{TEDNESSD}\\ \texttt{TERESTED}\end{pmatrix}\xrightarrow{\text{add}}\begin{pmatrix}\texttt{SDISINTER}\\ \texttt{EDNESSDIS}\\ \texttt{TEDNESSDI}\\ \texttt{TERESTEDN}\\ \texttt{NESSDISIN}\\ \texttt{RESTEDNES}\\ \texttt{SINTEREST}\\ \texttt{DISINTERE}\\ \texttt{DNESSDISI}\\ \texttt{INTERESTE}\\ \texttt{ERESTEDNE}\\ \texttt{SSDISINTE}\\ \texttt{ISINTERES}\\ \texttt{ESSDISINT}\\ \texttt{ESTEDNESS}\\ \texttt{STEDNESSD}\\ \texttt{NTERESTED}\end{pmatrix}\xrightarrow{\text{sort}}\begin{pmatrix}\texttt{DISINTERE}\\ \texttt{DNESSDISI}\\ \texttt{EDNESSDIS}\\ \texttt{ERESTEDNE}\\ \texttt{ESSDISINT}\\ \texttt{ESTEDNESS}\\ \texttt{INTERESTE}\\ \texttt{ISINTERES}\\ \texttt{NESSDISIN}\\ \texttt{NTERESTED}\\ \texttt{RESTEDNES}\\ \texttt{SDISINTER}\\ \texttt{SINTEREST}\\ \texttt{SSDISINTE}\\ \texttt{STEDNESSD}\\ \texttt{TEDNESSDI}\\ \texttt{TERESTEDN}\end{pmatrix}$$

Atkodēšanas procedūru varētu turpināt, bet ievērojam, ka šajā brīdī jau atkodēti 9 simboli no 17, turklāt atkodējamais vārds ir leksikogrāfiski pirmais. Tādēļ arī, pierakstot atlikušos simbolus, tas sāksies ar prefiksu `DISINTERE`. Ievērojam arī, ka pirms burtiem `DIS` cikliskajās permutācijās vienmēr parādās burti `SS`. Tātad atkodējamā virkne beidzas ar `SS`. Vienīgā šāda virkne ir sestajā rindiņā. Abas virknes sarakstot kopā, iegūstam meklēto vārdu:

$$\overbrace{\texttt{DISINTERE}\underbrace{\texttt{STEDNESS}}}$$

5.uzdevums (2+3 punkti): Kvantizācijas algoritms saņem ieejā attēla krāsu intensitātes, kuras ir veseli skaitļi $x \in \{0, 1, 2, \ldots, 255\}$ un atgriež apakšējo veselo daļu: $\left\lfloor \frac{x}{10} \right\rfloor$ jeb nomet tā decimālpieraksta pēdējo ciparu.

(a) Ja visas šīs kvantizētās vērtības būtu jāsūta, izmantojot vienādu, fiksētu bitu skaitu – cik lielu saspiešanas attiecību (*compression ratio* – sākotnējā faila izmēra attiecību pret saspiestā faila izmēru) varētu sasniegt?

(b) Pieņemot, ka visas krāsu intensitātes (no 0 līdz 255) ir ar vienādām varbūtībām, kāda ir jaunās, kvantizētās ziņojumu virknes entropija? Kāda būtu teorētiski labākā iespējamā saspiešanas attiecība (ja izmantotu aritmētisko kodējumu vai citu optimālu metodi, kas tuvojas entropijas noteiktajai saspiešanas robežai).

6.uzdevums (4 punkti): Spēlētājs X vēlas nosūtīt spēlētājam Y dažus ceturtās pakāpes polinomus ar veseliem koeficientiem:

$$f(x) = a_0x^4 + a_1x^3 + a_2x^2 + a_3x + a_4, \quad \text{kur} \quad a_0, \ldots, a_4 \in \mathbb{Z}.$$

Polinoma koeficientu a_i vietā spēlētājs X sūta s vērtības dažādiem veseliem argumentiem:

$$f(0), f(1), \ldots, f(s-1).$$

Vidū starp spēlētājiem X un Y atrodas ļaunprātīgais Šlopsterklopsters, kurš ne vairāk kā piecas no visām s nosūtītajām vērtībām drīkst nomainīt ar citiem skaitļiem (bet drīkst nomainīt arī mazāku skaitu vērtību vai nenomainīt nevienu).

Uzrakstīt nevienādību attiecībā pret parametru s (un atrisināt to), lai uzzinātu mazāko polinoma vērtību skaitu s, kuram spēlētājs Y noteikti varēs atjaunot X'a sūtīto polinomu, lai kā arī nerīkotos Šlopsterklopsters.

7.uzdevums (2+2+2 punkti): (Ieteikums: Šajā uzdevumā var izmantot algebriskas identitātes par skaitļu kāpināšanu, Eilera teorēmu u.c. skaitļu teorijas rezultātus.)

(a) Kriptogrāfijas algoritmam ir jāaprēķina $4^{143} \pmod{199}$ (atlikums, kas rodas dalot 4^{139} ar 199). Algoritms var ielūkoties reizināšanas tabulā pēc moduļa 199: ievadīt divus skaitļus a, b un saņemt to reizinājumu $a \cdot b \pmod{199}$ (ab atlikumu, dalot ar 199). Kādu mazāko reižu skaitu pietiek ielūkoties reizināšanas tabulā, lai atrastu $4^{139} \pmod{199}$.
Piezīme. Algoritms var veikt arī dalīšanu ar atlikumu un atņemšanu, bet jāsaskaita tikai reizināšanas darbības. (Acīmredzot, pietiek veikt 142 reizināšanas; Jūsu uzdevums ir reizināšanu skaitu pēc iespējas samazināt.)

(b) Kriptogrāfijas algoritmam ir jāaprēķina $4^{143} \pmod{7}$. Algoritms var ielūkoties reizināšanas tabulā pēc moduļa 7. Kādu mazāko reizināšanu skaitu vajag, lai atrastu rezultātu?

(c) Atrast $4^{143} \pmod{7}$ vērtību; parādīt, kā tā iegūta.

Atrisinājums:

(a)

(b)

(c) Pēc Eilera teorēmas, katram skaitlim a, kas nedalās ar 7, ir spēkā $a^6 \equiv 1 \pmod{7}$. Tai skaitā $4^6 \equiv 1 \pmod{7}$:

$$4^{138+5} = 4^{6\cdot 23+5} = (4^6)^{23} \cdot 4^5 \equiv 1^{23} \cdot 4^5 = 4^3 \cdot 4^2 = 64 \cdot 16.$$

64 atlikums, dalot ar 7 ir 1, bet 16 atlikums, dalot ar 7 ir 2. To reizinājums ir $1 \cdot 2 = 2$.