

Beigu eksāmens

Termins: 2020.gada 21.decembris, līdz vakaram
Iesniegšanas veids: E-studiju vide.

1.uzdevums (Rabina-Karpa algoritms)

RNA vīrusu genomu virknes pieraksta ar burtiem A, C, G, U. Mums dots garš teksts $T[0..n-1]$ garumā n , kas pierakstīts ar šiem burtiem. Un tajā jāmeklē paraudzīš, kas norāda uz vīrusa mutāciju: GUCAGA.

(A) Alise aizstāj šos četrus RNA genoma burtus ar to skaitliskajām vērtībām: A = 0, C = 1, G = 2, U = 3. Tā kā vīrusam-mutantam atbilstošais paraugs ir garumā 6 simboli, arī meklējamais logs ir tikpat garš. Katrai nobīdei $s \in [0, n-6]$ aplūkojam pārbaudāmajā tekstā T kaut kādus sešus pēc kārtas esošus simbolus $T[s]$, $T[s+1]$, $T[s+2]$, $T[s+3]$, $T[s+4]$, $T[s+5]$. Visi tie ir no kopas $\{0, 1, 2, 3\}$. Rēķinām hešfunkciju:

$$h_A(T[s..s+5]) = \left(\sum_{k=0}^5 T[s+k] \cdot x^k \right) \pmod{p},$$

kur polinoma mainīgais $x = 4$, bet pirmskaitlis $p = 1093$ (t.i. pirmskaitlis, kurš nedaudz lielāks par $2^5 = 1024$).

Atrast, cik daudzām 6-burtu kombinācijām w , kas uzrakstāmas ar alfabētu A, C, G, U būs spēkā kolīzija ar meklējamo paraugu:

$$h_A(w) = h_A(\text{GUCAGA}).$$

(B) Bobs izmanto citu hešfunkciju: Rabina-Karpa algoritma autora ieteikto Rabina digitālnospiedumu (Rabin fingerprint), sk. <https://bit.ly/2LUwuxo>. Viņš iekodē 6 pēc kārtas sekojošus RNA alfabēta burtus no meklējamā teksta T par 12 bitu virknīti (A = 00, C = 01, G = 10, U = 11). Tad pieraksta to kā 11.pakāpes polinomu ar 12 koeficientiem:

$$f(x) = m_0 + m_1x + \dots + m_{11}x^{11},$$

kur (atšķirībā no Alises hešfunkcijas h_A) pirmais burts dod polinomā jaunākos locekļus, bet pēdējais burts dod vecākos. Pēc tam Bobs dala iegūto 11.pakāpes polinomu ar nereducējamu 10.pakāpes polinomu $Q(x) = x^{10} + x^3 + 1$ koeficientu laukā $GF(2)$ (t.i. pēc pirmskaitļa 2 moduļa) un iegūst atlikumu $R(x)$, kas ir Boba hešfunkcijas vērtība. Piemēram, mutantu vīrusa raksturīgajai virknītei GUCAGA atbilst bitu virknīte 10.11.01.00.10.00. No tās rodas polinoms:

$$P(x) = 1 + 1x^2 + 1x^3 + 1x^5 + 1x^8.$$

Šī polinoma dalījums ar $x^{10} + x^3 + 1$ dod atlikumu, kas ir viņš pats (bet tiek jau aplūkots kā

9.pakāpes polinoms, nevis 11.pakāpes polinoms), t.i. Boba hešfunkcija:

$$h_B(\text{GUCAGA}) = 10.11.01.00.10.$$

Kā redzam, pēc hešfunkcijas pēdējie divi biti “pazūd”. Atrast, cik daudzām 6-burtu kombinācijām w , kas uzrakstāmas ar alfabētu A, C, G, U būs spēkā kolīzija ar meklējamo paraugu:

$$h_B(w) = h_B(\text{GUCAGA}).$$

Piezīme. Boba gadījumā (atšķirībā no Alises) polinomu $P(x)$ izmanto nevis, lai aprēķinātu polinoma vērtību kādai mainīgā x vērtībai, bet gan kā simbolisku pierakstu, lai iegūtu polinomu dalījuma atlikumu. Lai redzētu, kā darbojas polinomu aritmētika pēc 2 moduļa, minēsim vēl vienu Boba hešfunkcijas aprēķina piemēru. Ar Boba hešfunkciju iekodējamais 6-burtu vārds $w = \text{CAGUAU}$. Pārveidojam par bitu virknīti: 01.00.10.11.00.11 un uzrakstām sākotnējo 11.pakāpes polinomu:

$$\begin{aligned} P(x) &= \\ &= 0 + 1x^1 + 0x^2 + 0x^3 + 1x^4 + 0x^5 + 1x^6 + 0x^7 + 0x^8 + 0x^9 + 1x^{10} + 1x^{11} = \\ &= x + x^4 + x^6 + x^7 + x^{10} + x^{11} = \\ &= x^{11} + x^{10} + x^7 + x^6 + x^4 + x. \end{aligned}$$

Lai atrastu atlikumu, dalot ar $Q(x) = x^{10} + x^3 + 1$, vispirms atrodam $P(x) - x \cdot Q(x)$, lai atbrīvotos no saskaitāmā x^{11} :

$$\begin{aligned} P(x) - x \cdot Q(x) &= \\ &= (x^{11} + x^{10} + x^7 + x^6 + x^4 + x) - x \cdot (x^{10} + x^3 + 1) = \\ &= x^{11} + x^{10} + x^7 + x^6 + x^4 + x - x^{11} - x^4 - x = \\ &= x^{10} + x^7 + x^6. \end{aligned}$$

Tagad atņemam $Q(x)$, lai atbrīvotos arī no saskaitāmā x^{10} :

$$\begin{aligned} (x^{10} + x^7 + x^6) - (x^{10} + x^3 + 1) &= \\ &= x^7 + x^6 - x^3 - 1 = x^7 + x^6 + x^3 + 1. \end{aligned}$$

Šajos pārveidojumos izmantojam, ka $-1 \equiv 1 \pmod{2}$. Tātad $R(x) = x^7 + x^6 + x^3 + 1$ arī ir meklētais atlikums. Pārrakstām to kā 10-bitu virknīti, sākot no jaunākā koeficienta:

$$R(x) = 1 + x^3 + x^6 + x^7 = \sum_{i=0}^9 a_i \cdot x^i.$$

Iegūstam $(a_0, \dots, a_9) = (1, 0, 0, 1, 0, 0, 1, 1, 0, 0)$ un tātad Boba hešfunkcija

$$h_B(\text{CAGUAU}) = 10.01.00.11.00.$$

Kā redzam, visos šajos pārveidojumos mums nav jāievieto konkrētas x vērtības; Boba gadījumā x ir tikai simbolisks apzīmējums, kas palīdz veikt polinomu dalīšanu ar atlikumu. Mūs interesējošais rezultāts pats ir polinoms $R(x)$.

(C) Alises un Boba hešfunkcijām atrast varbūtību, ka Rabina-Karpa algoritms sastaps meklējamā tekstā kolīziju, ja meklējamais 6 burtu paraudzīš P ar vienādu varbūtību ir jebkura 6 burtu RNA virkne, bet teksts T ir nejauši veidots un garš.

2.uzdevums (Garākais palindroms).

Mūsu uzdevums ir atrast garāko substringu dotajā stringā P , kas vienlaikus būtu palindroms (lasītos no abiem galiem vienādi). Ja šādu garāko palindromu ir vairāki, pietiek atrast vienu no tiem. Piemēram, vārdā **BANANA** garākais palindroms ir ANANA, vārdā **ANNA** garākais palindroms ir pats ANNA, vārdā **ABRAKADABRA** garākais palindroms ir AKA, bet vārdā **ABCD** tas ir viena burtu strings, piemēram, A.

(A) Aplūkosim naīvo algoritmu, kas apskata visus iespējamās dotā stringa P apakšstringus; katram no tiem pārbauda, vai tas ir palindroms. Kāda ir šī algoritma laika sarežģītība $O(f(n))$? (Šeit n - ievades stringa garums.)

(B) Kāds programmētājs piedāvā lietot Ukkona algoritmu un izveidot sufiksu koku, kurš uzbūvēts no sekojošu divu vārdu sufiksiem:

$$P = \text{BANANAS\$}, P_{rev} = \text{ANANAB\#}.$$

P_{rev} ir P uzrakstīts no otra gala un izmantoti divi dažādi beigu marķieri \$ (sākotnējam stringam) un # (reversajam stringam).

Izveidotajā kokā atrodam visdziļāk esošo iekšējo virsotni, zem kuras ir gan zilas, gan zaļas lapas (t.i. kas var beigties gan ar #, gan ar \$). Mūsu gadījumā šī virsotne ir **ANANA** (apvilīts ar aplīti Attēlā 1. Tas ir arī garākais palindroms, kas ietilpst vārdā **BANANA**

Piezīme. Virsotnes v dziļumu sufiksu kokā definē kā burtu skaitu, kas jānolasa, lai no sufiksu koka saknes tiktu līdz v .

šējā virsotne, kuru var pabeigt gan kā stringa P sufiksu, gan kā P_{rev} sufiksu, nemaz nav palindroms (vai arī nav visgarākais starp palindromiem, kurš ietilpst stringā P).

(D) Aprakstīt tādu palindromu meklēšanas metodi, kas arī var izmantot Attēlam 1 līdzīgu P un P_{rev} kopīgo sufiksu koku, bet tam nemēdz būt pretpiemēri (kā **(C)**). Atrast Jūsu palindromu meklēšanas metodei laika sarežģītību. (Velams, lai tā strādātu ātrāk nekā algoritms no **(A)**.)

3.uzdevums (Primārais un duālais LP)

Dots primārais LP uzdevums: Maksimizēt $z = 2x_1 + 5x_2$, kur $3x_1 + 7x_2 = 12$ un $x_1, x_2 \geq 0$.

(A) Kāds ir primārā LP mērķfunkcijas $2x_1 + 5x_2$ maksimums, un pie kuriem x_i to var sasniegt.

(B) Formulēt dotajam primārajam duālo LP uzdevumu.

(C) Atrast duālā uzdevuma mērķfunkcijas minimumu un kādām mainīgo vērtībām to sasniedz.

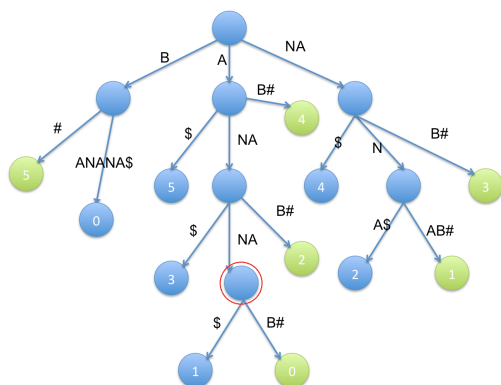


Figure 1: Sufiksu koks 2 stringiem.

Pēc Attēla 1 parauga izveidot un uzzīmēt kopīgu sufiksu koku stringiem

$$P = \text{KLIBIBIKLI\$} \text{ un } P_{ren} = \text{ILKIBIBILK\#}.$$

(C) Atrast pretpiemēru iepriekšējā punktā aprakstītajai palindromu meklēšanas metodei, kur P un P_{rev} kopīgajā sufiksu kokā atrastā dzilākā iek-