

Șiruri audioactive

Răzvan-Anton Mureanu <razvan.mureanu@cnmbct.ro>

Ștefan Patrichi <stefan.patrichi.07@cnmbct.ro>

Colegiul Național „Mircea cel Bătrân” Constanța

$AI + \alpha + Z = \text{Matematica noilor generații}$

8 noiembrie 2025

Care este următorul termen?

1

11

21

1211

111221

312211

13112221

1113213211

Care este următorul termen?

111221

312211

Care este următorul termen?

111221
31 22 11

- Alfabet: $\Sigma = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Cuvânt: $\alpha = a_1 a_2 \dots a_k$ cu $a_i \in \Sigma$
- Lungimea cuvântului: $|\alpha| = k$
- Concatenarea: $\alpha = a_1 a_2 \dots a_k, \beta = b_1 b_2 \dots b_l$
Atunci $\alpha \cdot \beta = \alpha\beta = a_1 a_2 \dots a_k b_1 b_2 \dots b_l$
- Cuvânt vid: ε cu $|\varepsilon| = 0$ și $\alpha\varepsilon = \varepsilon\alpha = \alpha \quad \forall \alpha$
- Mulțimea cuvintelor:
 $\Sigma^* = \{a_1 a_2 \dots a_k \mid a_i \in \Sigma, k \in \mathbb{N}\}$
- $(\Sigma^*, \cdot) = \textbf{monoidul liber}$ generat de mulțimea Σ

Convenții pentru notația multiplicativă

- Notăție: $\underbrace{aa \dots a}_{\text{de } m \text{ ori}} \underbrace{bb \dots b}_{\text{de } n \text{ ori}} = a^m b^n$
- $a^m a^n = a^{m+n}$

Convenții pentru notația multiplicativă

- Notatie: $\underbrace{aa \dots a}_{\text{de } m \text{ ori}} \underbrace{bb \dots b}_{\text{de } n \text{ ori}} = a^m b^n$
- $a^m a^n = a^{m+n}$
- Exemplu: $111221 = 1^3 2^2 1^1 \rightarrow 312211$
- Exemplu: $33333333333 = 3^{11} \rightarrow 113$

Definiție.

Un șir care evoluează după regula descrisă anterior se numește *șir look-and-say* (*privește-și-spune*).

Definiție.

Primul termen al unui șir look-and-say se numește *origine*.

Definiție.

Un șir look-and-say a cărui origine nu conține nicio înșiruire de mai mult de 9 cifre identice adiacente se numește *șir compact*.

Definiție.

Notăm cu $\mathcal{C} \subset \Sigma^*$ mulțimea tuturor cuvintelor care apar în șiruri compacte, excluzând originea.

Definiție.

Un șir look-and-say a cărui origine:

- (a) nu conține nicio cifră mai mare decât 3 și
- (b) nu conține nicio înșiruire de mai mult de 3 cifre identice adiacente

se numește *șir armonic*.

Definiție.

Notăm cu $\mathcal{A} \subset \mathcal{C} \subset \Sigma^*$ mulțimea tuturor cuvintelor care apar în șiruri armonice, excluzând originea.

- $LS : \mathcal{C} \rightarrow \mathcal{C}, \boxed{LS(a_1^{n_1} a_2^{n_2} \dots a_k^{n_k}) = n_1 a_1 n_2 a_2 \dots n_k a_k}$

- $LS : \mathcal{C} \rightarrow \mathcal{C}$, $LS(a_1^{n_1} a_2^{n_2} \dots a_k^{n_k}) = n_1 a_1 n_2 a_2 \dots n_k a_k$
- Condiție: $a_i \neq a_{i+1}$ ($a_i^m a_i^n = a_i^{m+n}$)
- $LS(777777) = 67$, nu 2747 sau 5717.

Propoziție.

Mulțimea \mathcal{C} coincide cu mulțimea cuvintelor $a_1a_2 \dots a_k$ pentru care:

- (a) $2 \mid k$,
- (b) $a_{2i} \neq a_{2i+2}$.

Proprietăți de bază

Propoziție.

Mulțimea \mathcal{C} coincide cu mulțimea cuvintelor $a_1a_2 \dots a_k$ pentru care:

- (a) $2 \mid k$,
- (b) $a_{2i} \neq a_{2i+2}$.

Demonstrație.

Pentru incluziunea \supset , fie $a_1b_1a_2b_2 \dots a_kb_k$ un cuvânt astfel încât $b_i \neq b_{i+1}$. Atunci putem să-i construim inversul, $b_1^{a_1}b_2^{a_2} \dots b_k^{a_k}$. □

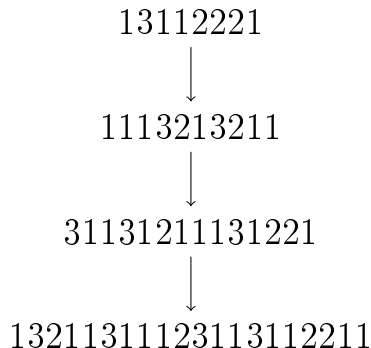
Observație.

Am demonstrat că funcția f este bijectivă!

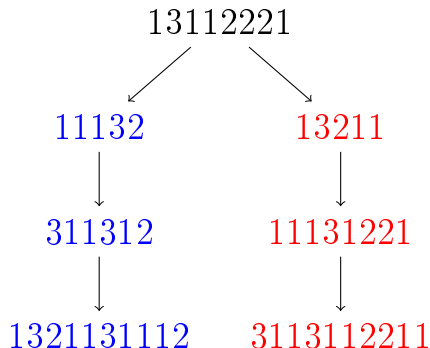
Propoziție.

Mulțimea \mathcal{A} coincide cu mulțimea cuvintelor $a_1a_2 \dots a_k$ pentru care:

- (a) $2 \mid k$,
- (b) $a_{2i} \neq a_{2i+2}$ și
- (c) $a_i \in \{1, 2, 3\}$.



Proprietăți de bază – Descompunerea



- $LS^n(S \cdot D) = LS^n(S) \cdot LS^n(D)$
- Ultima cifră a lui $LS^n(S) \neq$ prima cifră a lui $LS^n(D)$, pentru $\forall n \geq n_0$.

- $LS^n(S \cdot D) = LS^n(S) \cdot LS^n(D)$
- Ultima cifră a lui $LS^n(S) \neq$ prima cifră a lui $LS^n(D)$, pentru $\forall n \geq n_0$.
- Dacă y se descompune în x_1, x_2, \dots, x_k , atunci:

$$LS^n(y) = \prod_{i=1}^k LS^n(x_i)$$

$$|LS^n(y)| = \sum_{i=1}^k |LS^n(x_i)|$$

Rezultate:

- Modul în care se repetă începutul și sfârșitul termenilor dintr-un șir look-and-say



John Conway (1937-2020)

Rezultate:

- Condiții necesare și suficiente pentru descompunere



John Conway (1937-2020)

Rezultate:

- Teoremă: **Există 92 de** *elemente atomice* astfel încât orice origine se descompune eventual într-o succesiune de *elemente atomice*



John Conway (1937-2020)

Rezultate:

- Rata asimptotică de creștere a lungimii cuvintelor dintr-un șir look-and-say,

$$\lambda = \lim_{n \rightarrow \infty} \frac{|s_{n+1}|}{|s_n|}$$



John Conway (1937-2020)

Metodă de calcul pentru λ

- e_i = cele d elemente atomice
- t_{ij} = de câte ori apare e_i în regula de evoluție a lui e_j
- f_i = de câte ori apare e_i în descompunerea originii șirului

#	Subsequence	Length	Evolves Into
1	1112	4	(63)
2	1112133	7	(64)(62)
3	111213322112	12	(65)
4	111213322113	12	(66)
5	1113	4	(68)
6	11131	5	(69)
7	111311222112	12	(84)(55)
8	111312	6	(70)
9	11131221	8	(71)
10	1113122112	10	(76)
11	1113122113	10	(77)
12	11131221131112	14	(82)
13	111312211312	12	(78)
14	11131221131211	14	(79)
15	111312211312113211	18	(80)
16	111312211312113221133211322112211213322112	42	(81)(29)(91)
17	111312211312113221133211322112211213322113	42	(81)(29)(90)
18	11131221131211322113322112	26	(81)(30)
19	11131221133112	14	(75)(29)(92)
20	1113122113322113111221131221	28	(75)(32)
21	11131221222112	14	(72)
22	111312212221121123222112	24	(73)
23	111312212221121123222113	24	(74)
24	11132	5	(83)
25	1113222	7	(86)
26	1113222112	10	(87)
27	1113222113	10	(88)
28	11133112	8	(89)(92)

Fragment din *Tabelul periodic* al celor
 $d = 92$ elemente

- Fie n_0 astfel încât $LS^{n_0}(s_0) = s_{n_0}$ se descompune în elemente atomice.

- Fie n_0 astfel încât $LS^{n_0}(s_0) = s_{n_0}$ se descompune în elemente atomice.

$$|s_{n_0+1}| = \sum_{i=1}^d (\text{nr. apariții } e_i \text{ în } LS(s_{n_0})) \cdot |e_i|$$

Metodă de calcul pentru λ

- Fie n_0 astfel încât $LS^{n_0}(s_0) = s_{n_0}$ se descompune în elemente atomice.

$$\begin{aligned}|s_{n_0+1}| &= \sum_{i=1}^d (\text{nr. apariții } e_i \text{ în } LS(s_{n_0})) \cdot |e_i| \\ &= \sum_{i=1}^d \left(\sum_{j=1}^d t_{ij} f_j \right) \cdot |e_i|\end{aligned}$$

Metodă de calcul pentru λ

- Fie n_0 astfel încât $LS^{n_0}(s_0) = s_{n_0}$ se descompune în elemente atomice.

$$\begin{aligned}|s_{n_0+1}| &= \sum_{i=1}^d (\text{nr. apariții } e_i \text{ în } LS(s_{n_0})) \cdot |e_i| \\ &= \sum_{i=1}^d \left(\sum_{j=1}^d t_{ij} f_j \right) \cdot |e_i|\end{aligned}$$

Metodă de calcul pentru λ

- Fie n_0 astfel încât $LS^{n_0}(s_0) = s_{n_0}$ se descompune în elemente atomice.

$$\begin{aligned}|s_{n_0+1}| &= \sum_{i=1}^d (\text{nr. apariții } e_i \text{ în } LS(s_{n_0})) \cdot |e_i| \\ &= \sum_{i=1}^d \left(\sum_{j=1}^d t_{ij} f_j \right) \cdot |e_i|\end{aligned}$$

$$\mathbf{T} = \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1d} \\ t_{21} & t_{22} & \cdots & t_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ t_{d1} & t_{d2} & \cdots & t_{dd} \end{pmatrix} \quad \mathbf{f} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_d \end{pmatrix} \quad \mathbf{l} = \begin{pmatrix} |e_1| \\ |e_2| \\ \vdots \\ |e_d| \end{pmatrix}$$

Metodă de calcul pentru λ

- Fie n_0 astfel încât $LS^{n_0}(s_0) = s_{n_0}$ se descompune în elemente atomice.

$$\begin{aligned}|s_{n_0+1}| &= \sum_{i=1}^d (\text{nr. apariții } e_i \text{ în } LS(s_{n_0})) \cdot |e_i| \\ &= (\mathbf{T}\mathbf{f}) \cdot \mathbf{1} = \mathbf{1}^\top (\mathbf{T}\mathbf{f})\end{aligned}$$

$$\mathbf{T} = \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1d} \\ t_{21} & t_{22} & \cdots & t_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ t_{d1} & t_{d2} & \cdots & t_{dd} \end{pmatrix} \quad \mathbf{f} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_d \end{pmatrix} \quad \mathbf{1} = \begin{pmatrix} |e_1| \\ |e_2| \\ \vdots \\ |e_d| \end{pmatrix}$$

- Fie n_0 astfel încât $LS^{n_0}(s_0) = s_{n_0}$ se descompune în elemente atomice.

$$\begin{aligned}|s_{n_0+1}| &= \sum_{i=1}^d (\text{nr. apariții } e_i \text{ în } LS(s_{n_0})) \cdot |e_i| \\ &= (\mathbf{Tf}) \cdot \mathbf{1} = \mathbf{1}^\top (\mathbf{Tf})\end{aligned}$$

- Analog,

$$|s_{n_0+k}| = (\mathbf{T}^k \mathbf{f}) \cdot \mathbf{1} = \mathbf{1}^\top (\mathbf{T}^k \mathbf{f})$$

Metodă de calcul pentru λ

- λ_i, \mathbf{v}_i = valori/vectori proprii ai matricei \mathbf{T}
- λ_p, \mathbf{v}_p = valoare/vector propriu Perron a matricei \mathbf{T}

Metodă de calcul pentru λ

- λ_i, \mathbf{v}_i = valori/vectori proprii ai matricei \mathbf{T}
- λ_p, \mathbf{v}_p = valoare/vector propriu Perron a matricei \mathbf{T}
- Pentru simplitate, $n_0 = 0$.

Metodă de calcul pentru λ

- λ_i, \mathbf{v}_i = valori/vectori proprii ai matricei \mathbf{T}
- λ_p, \mathbf{v}_p = valoare/vector propriu Perron a matricei \mathbf{T}
- Pentru simplitate, $n_0 = 0$.
- Când \mathbf{T} e diagonalizabilă,

$$\frac{1}{\lambda^n} \mathbf{T}^n \mathbf{f} = \frac{1}{\lambda^n} \mathbf{T}^n \sum_{i=1}^d \alpha_i \mathbf{v}_i$$

Metodă de calcul pentru λ

- λ_i, \mathbf{v}_i = valori/vectori proprii ai matricei \mathbf{T}
- λ_p, \mathbf{v}_p = valoare/vector propriu Perron a matricei \mathbf{T}
- Pentru simplitate, $n_0 = 0$.
- Când \mathbf{T} e diagonalizabilă,

$$\begin{aligned}\frac{1}{\lambda^n} \mathbf{T}^n \mathbf{f} &= \frac{1}{\lambda^n} \mathbf{T}^n \sum_{i=1}^d \alpha_i \mathbf{v}_i \\ &= \sum_{i=1}^d \alpha_i \left(\frac{\lambda_i}{\lambda} \right)^n \mathbf{v}_i \rightarrow \alpha_p \mathbf{v}_p\end{aligned}$$

- Acum,

$$\lim_{n \rightarrow \infty} \frac{|s_{n+1}|}{|s_n|} = \lim_{n \rightarrow \infty} \lambda_p \frac{\frac{1}{\lambda_p^{n+1}} \mathbf{T}^{n+1} \mathbf{f} \cdot \mathbf{1}}{\frac{1}{\lambda_p^n} \mathbf{T}^n \mathbf{f} \cdot \mathbf{1}}$$

- Acum,

$$\lim_{n \rightarrow \infty} \frac{|s_{n+1}|}{|s_n|} = \lim_{n \rightarrow \infty} \lambda_p \frac{\frac{1}{\lambda_p^{n+1}} \mathbf{T}^{n+1} \mathbf{f} \cdot \mathbf{l}}{\frac{1}{\lambda_p^n} \mathbf{T}^n \mathbf{f} \cdot \mathbf{l}}$$

- Acum,

$$\begin{aligned}\lim_{n \rightarrow \infty} \frac{|s_{n+1}|}{|s_n|} &= \lim_{n \rightarrow \infty} \lambda_p \frac{\frac{1}{\lambda_p^{n+1}} \mathbf{T}^{n+1} \mathbf{f} \cdot \mathbf{l}}{\frac{1}{\lambda_p^n} \mathbf{T}^n \mathbf{f} \cdot \mathbf{l}} \\ &= \lambda_p \frac{\alpha_p \mathbf{v}_p \cdot \mathbf{l}}{\alpha_p \mathbf{v}_p \cdot \mathbf{l}} \\ &= \lambda_p \approx 1.303577269034 \dots\end{aligned}$$

Varianțiuni de șiruri look-and-say

Varianțiuni de șiruri look-and-say

- LS cu nr. de apariții postpus (LS_{post})

$$LS_{\text{post}}(a_1^{n_1} a_2^{n_2} \dots a_k^{n_k}) = a_1 n_1 a_2 n_2 \dots a_k n_k$$

- LS în baza 2 (LS_{b2})
- LS cu cifre romane (LS_{roman})
- LS cu suprimarea cifrelor *singure* (LS_{sng})

$$LS_{\text{sng}}(a_1^{n_1} a_2^{n_2} \dots a_k^{n_k}) = m_1 a_1 m_2 a_2 \dots m_k a_k,$$

unde:

$$m_i = \begin{cases} \varepsilon, & n_i = 1 \\ n_i, & n_i \geq 2 \end{cases}$$

- Elemente atomice
- Puncte fixe (de ex. ε și 22 pentru LS -ul obișnuit)
- Rata asimptotică de creștere a lungimii cuvintelor (λ)
- Stabilizarea (șirul (s_n) constant de la un rang încolo)
- Alte proprietăți specifice (regularități/ciclicități)

$LS : 1, 11, 21, 1211, 111221, 312211, 13112221, \dots$

$LS_{\text{post}} : 1, 11, 12, 1121, 122111, 112213, 12221131 \dots$

$LS : 1, 11, 21, 1211, 111221, 312211, 13112221, \dots$

$LS_{\text{post}} : 1, 11, 12, 1121, 122111, 112213, 12221131 \dots$

Teoremă.

$LS_{\text{post}}^n(s_0) = R(LS^n(s_0))$, unde $R : \mathcal{C} \rightarrow \mathcal{C}$ este operatorul de *răsturnare* a cuvintelor.

lată o demonstrație scurtă și curată.

Notăm:

- L = operatorul look-and-say **clasic** (numărul de apariții **urmat** de cifra citită).
- L_{post} = operatorul look-and-say cu numărul de apariții **postpus** (cifra citită urmată de număr).
- R = operatorul care răstoarnă (oglindește) un șir de cifre.
- $s_{n+1} = L(s_n)$ și $s'_{n+1} = L_{\text{post}}(s'_n)$, cu $s_0 = s'_0 = 1$.

Lemă-cheie

Pentru orice șir x format din cifre, avem

$$L_{\text{post}}(x) = R(L(R(x))).$$

Dovadă a lemei. Scriem x ca alternanță de „run-uri” de aceeași cifră:

$$x = a_1^{k_1} a_2^{k_2} \cdots a_m^{k_m}, \quad a_i \neq a_{i+1}.$$

Atunci

$$L(x) = k_1 a_1 k_2 a_2 \cdots k_m a_m.$$

Pe de altă parte, $R(x) = a_m^{k_m} \cdots a_1^{k_1}$, deci

$$L(R(x)) = k_m a_m \cdots k_1 a_1,$$

iar răsturnând din nou,

$$R(L(R(x))) = a_1 k_1 \cdots a_m k_m = L_{\text{post}}(x).$$



Lema e dovedită. \square

Inducție

Vrem să arătăm că pentru orice $n \in \mathbb{N}$,

$$s'_n = R(s_n).$$

- **Baza:** $n = 0$. Avem $s'_0 = 1 = R(1) = R(s_0)$.
- **Pas inductiv:** Presupunem $s'_n = R(s_n)$. Atunci, folosind lema,

$$s'_{n+1} = L_{\text{post}}(s'_n) = L_{\text{post}}(R(s_n)) = R(L(R(s_n))) = R(L(s_n)) = R(s_{n+1}).$$

Deci proprietatea se păstrează la pasul $n + 1$.

Prin inducție, $s'_n = R(s_n)$ pentru toate n .

(Mini-exemplu)

$s_0 = 1 \Rightarrow s_1 = L(1) = 11$, iar $s'_1 = L_{\text{post}}(1) = 11$; într-adevăr, $R(s_1) = R(11) = 11$.

Pentru $s_1 = 11$, $s_2 = L(11) = 21$; în varianta postpusă, $s'_2 = L_{\text{post}}(11) = 12$, iar $R(s_2) = R(21) = 12$.

Asfel, șirul cu numărul postpus este, la fiecare pas, răsturnatul șirului obișnuit. ✅



+ Ask anything



$LS_{b_2} : 1, 11, 101, 111011, 11110101, 100110111011$

Thought for 1m 3s >

Nu există puncte fixe nenule pentru funcția „look-and-say” în baza 2.

De ce

Fie cuvântul w și prima lui rună b^r (cu $b \in \{0, 1\}$, $r \geq 1$). În $LS_{b2}(w)$ această rună este înlocuită cu $\text{bin}(r) b$, care are lungime $\ell = \lfloor \log_2 r \rfloor + 1 + 1$.

Dacă w ar fi punct fix, atunci primele r simboluri din $LS_{b2}(w)$ ar trebui să fie tot b , deci trebuie să avem neapărat $\ell = r$. Singurele soluții ale $r = \lfloor \log_2 r \rfloor + 2$ sunt $r \in \{3, 4\}$.

- Pentru $r = 4$: $\text{bin}(4) b = 100 b$ nu poate fi egal cu $bbbb$. Contradicție.
- Pentru $r = 3$: $\text{bin}(3) b = 11 b$ este egal cu bbb doar dacă $b = 1$, deci prefixul lui w ar fi „111”.
Următoarea rună ar fi „0...”. Oricare ar fi lungimea ei s , pentru a menține punct fix trebuie din nou $s \in \{3, 4\}$, dar $\text{bin}(3) 0 = 110 \neq 000$ și $\text{bin}(4) 0 = 1000 \neq 0000$. Contradicție.

Prin urmare, niciun cuvânt binar nenul nu este fix. (Șirul vid nu este, în mod obișnuit, considerat termen valid.)





Thought for 4m 56s >

Mai jos dau o demonstrație (scurtă, dar completă) a faptului că

$$\lim_{n \rightarrow \infty} \frac{|b_{n+1}|}{|b_n|} = \rho, \quad \rho \approx 1,465571,$$

unde ρ este unica rădăcină reală a ecuației $x^3 - x^2 - 1 = 0$.

1) Definiție și o observație-cheie

Șirul „look-and-say în baza 2” pornește de la un cuvânt binar (de pildă „1”) și transformă fiecare **bloc maximal de 0-uri sau 1-uri** de lungime k în

(scrierea lui k în binar) \parallel (bitul blocului).

De exemplu: $1 \mapsto 11$ (un „1”), $11 \mapsto 101$ (doi „1”), $101 \mapsto 111011$ (un „1”, un „0”, un „1”), ș.a.m.d.

Lemă (lungimi de blocuri ≤ 3). Pentru toți $n \geq 2$, niciun b_n nu conține un bloc de patru (sau mai multe) zerouri ori de patru (sau mai multe) unu-uri.

Schiță de dovadă. Scrierea în binar a unui $k \geq 4$ are cel puțin două biți, iar concatenarea cu bitul blocului introduce alternanțe ce „taie” blocurile lungi; printr-o verificare pe cazuri (ce se propagă inductiv) se vede că nu poate apărea un bloc ≥ 4 . \square

Rezultă că toate blocurile care apar în oricare b_n (pentru n suficient de mare) au lungimi doar din

1. i
2. ii
3. iii
4. iiii
5. ivi
6. iiivii
7. iiiiviii
8. viiviii
9. iviiiiivivi
10. iiiviviiiviiivii
11. iiiiviiiviiiviiiviiivii
12. viiviiiiiviiiviiiviiivii

1. i
2. ii
3. iii
4. iiii
5. i vi
6. iii vii
7. iiii viii
8. vii viiii
9. i viiiivivi
10. iii vivii viii vii
11. iiii viii viiii viiiii viii
12. vii viiiii vivii vvi viiii

1. i
5. ivi
9. iviiiiviii
13. iviiiivviiiviiiiviiiiviiiiviii
17. iviiiivviiiiviiiiviiiivviiivviiiviiiiviiiivviiiviiiiviiiivivi
21. iviiiivviiiiviiiiviviiivviiiiviiiiviiiiviviiiviiiivviiiviviiivv...
25. iviiiivviiiiviiiivviiiviiiiviiiivviiivviiiiviiiiviiiiviiiiviii...
29. iviiiivviiiiviiiivviiiiviiiiviiiivviiivviiiiviiiiviiiivviiiviv...
33. iviiiivviiiiviiiivviiiiviiiiviviiivviiiiviiiiviiiivviiivviiiiv...

1. i
5. ivi
9. $iviiiiviii$
13. $iviiiivviiiviiiiviiiiviii$
17. $iviiiivviiiiviiiiviiiivviiivviiiviiiivviiiviiiiviiiivivi$
21. $iviiiivviiiiviiiiviviiivviiiiviiiivviiivviiivv...$
25. $iviiiivviiiiviiiivviiiviiiiviiiivviiivviiiiviiiiviiiiviii...$
29. $iviiiivviiiiviiiivviiiiviiiivviiivviiiiviiiivviiiviv...$
33. $iviiiivviiiiviiiivviiiiviiiivviiivviiiivviiivviiivv...$

Mai jos este o demonstrație „mecanică” (de tip automată finită) pentru formula:

$$k(n) = \# \text{blocuri comune la început între } s_n \text{ și } s_{n+4} = \left\lfloor \frac{n}{3} \right\rfloor + \delta(n),$$

cu excepția micii faze de pornire $n \leq 11$, iar pentru $n \geq 12$ corecția $\delta(n)$ este periodică de perioadă 12:

$$\delta(n \bmod 12) = (0, 2, 1, 0, 1, 2, 0, 1, 1, 1, 1, 1).$$

Fie (s_n) un șir look-and-say cu cifre romane, cu $s_{n+1} = LS(s_n)$. De exemplu, pentru $s_0 = i$, șirul este: $i, ii, iii, iiiv, vii, viiiv, viiiviii, viiiviii, viiiviii$ etc. Spunem că z se descompune în x și y și scriem $z = x.y$ dacă $LS^n(z) = LS^n(x) LS^n(y)$ pentru orice n natural. O mulțime A , finită, de cuvinte cu elemente din alfabetul $\Sigma = \{i, v\}$, se numește mulțime de elemente atomice dacă pentru orice valoare a lui s_0 există un număr natural n astfel încât $s_n = LS^n(s_0)$ se descompune în elemente din mulțimea A . Găsește mulțimea de elemente atomice a șirurilor look-and-say cu cifre romane.

Pe scurt: nu există puncte fixe nebanale. (Eventual, doar şirul vid, dacă îl permitem.)

De ce

Definim LS astfel: pentru fiecare bloc maximal a^k (aceeaşi literă romană $a \in \{i, v, x, l, c, d, m\}$ repetată de k ori), îl înlocuim cu $R(k) a$, unde $R(k)$ este scrierea lui k în cifre romane (cu notaţia standard, inclusiv formele subtractive).

Presupune că $s \neq \varepsilon$ este un punct fix: $LS(s) = s$.

Fie prima rună din s egală cu a^k (cu $k \geq 1$). Atunci primul bloc produs de LS este tocmai $R(k) a$. Dar cum $s = LS(s)$, prefixul de lungime k al lui s este simultan:

- a^k (pentru că e prima rună a lui s),
- şi $R(k) a$ (pentru că e primul bloc al lui $LS(s)$).

Rezultă

$$R(k) a = a^k \Rightarrow R(k) = a^{k-1}.$$

Deci reprezentarea romană a lui k ar trebui să fie o putere a *aceleiaşi* litere a , cu exact $k - 1$ caractere.

Dar asta e imposibil:

- Dacă $a = i$, atunci $R(k) \in \{i, ii, iii, iv, \dots\}$. Ca să fie $R(k) = i^{k-1}$, ar trebui $R(k)$ să fie format doar din i -uri; singurele cazuri sunt $k = 1, 2, 3$ cu $R(k) = i, ii, iii$, însă acestea au lungimi $1, 2, 3 \neq k - 1$.
- Dacă $a \in \{v, x, l, c, d, m\}$, singurele valori ale lui k pentru care $R(k)$ foloseşte doar litera a sunt $k \in \{5, 10, 20, 30, 50, 100, 200, 300, 500, 1000, 2000, 3000\}$. În toate aceste cazuri $|R(k)| \in \{1, 2, 3\}$, deci nu poate fi $|R(k)| = k - 1$.

Contradicţie. Aşadar nu există şir nevid cu $LS(s) = s$.

LS_{sng} : 333221, 33221, 23221, 23221, 23221, ...

Notăm cu $R(w)$ lungimea maximă a unui bloc (run) de cifre egale din w .

Lemma 1 (construcția blocurilor lungi)

Pentru orice $r \geq 3$, dacă în w apare un bloc a^r (cu $a \in \Sigma$), atunci în $T(w)$ toate blocurile rezultate din acest bloc au lungime maximă $\leq A(r) + 1$, unde $A(r)$ este numărul maxim de cifre egale consecutive în $\text{repr}(r)$. În particular,

$$A(r) + 1 \leq \lfloor \log_{10} r \rfloor + 2 \leq r - 1 \quad (\text{pentru } r \geq 3).$$

Prin urmare, dacă $R(w) \geq 3$, atunci $R(T(w)) \leq R(w) - 1$.

Demonstrație. Blocul a^r devine $\text{repr}(r) a$. Singurele blocuri de cifre egale din această imagine sunt (i) blocurile din interiorul $\text{repr}(r)$, a căror lungime maximă este $A(r)$, și (ii) eventual unirea ultimei cifre din $\text{repr}(r)$ cu a , ceea ce mai poate crește cu cel mult 1. Inegalitatea $A(r) + 1 \leq r - 1$ pentru $r \geq 3$ rezultă din $A(r) \leq \# \text{cifre}(r) = \lfloor \log_{10} r \rfloor + 1 \leq r - 2$. ■

Consecință. Aplicând Lemma 1 iterativ, după un număr finit de pași obținem un rang N cu

$$R(T^N(w)) \leq 2.$$

Lemma 2 (stabilizarea după ce toate blocurile au lungime ≤ 2)

Fie u un șir cu $R(u) \leq 2$. Atunci $R(T(u)) \leq 2$, iar toate blocurile de lungime 2 din $T(u)$ sunt exact „22”. Mai mult, $T(T(u)) = T(u)$; adică $T(u)$ este punct fix.

Vă mulțumim!