Určeno pro MUNDUS SYMBOLICUS 29 (2021)

Některé méně obvyklé poziční číselné soustavy

1. Věta o dělení se zbytkem

Zápis celých čísel v pozičních číselných soustavách se opírá o následující větu o dělení se zbytkem:

Věta 1. *Nechť n a g jsou celá čísla,* $g \geq 2$. *Pak existuje právě jedna taková dvojice celých čísel q, r, že*

$$n = q.g + r \tag{1}$$

a

$$r \in \{0, 1, \ldots g - 1\}. \tag{2}$$

Přitom platí:

Je-li $n > 0$, pak

$$n > q \geq 0 \tag{3}$$

Pro $n = 0$ je

$$q = r = 0 \tag{4}$$

Pro $n < 0$ platí

$$|n| \geq |q| \geq 1 \tag{5}$$

Pomocí rovnosti (1) a podmínky (2) definujeme dvě funkce div a rem (celočíselný podíl, resp. zbytek při dělení) dvou celočíselných proměnných n, g pro libovolné n a pro $g \geq 2$:

$$q = \mathrm{div}(n, g), \tag{6}$$

$$r = \mathrm{rem}(n, g) \tag{7}$$

K zápisu nezáporných celých čísel v ***klasické* poziční číselné soustavě o základu *g*** (v g-adické číselné soustavě) potřebujeme g číslic vyjadřujících čísla 0, 1, ..., $g - 1$. Každé nezáporné celé číslo n můžeme vyjádřit ve tvaru konečného součtu

$$n = \Sigma_{i=0}^{k} a_i g^i, \qquad (8)$$

kde $a_k \in \{0, 1, \ldots, g - 1\}$ jsou číslice, číslo n zapisujeme pomocí (k+1)-prvkové posloupnosti číslic.

$$n = a_k a_{k-1} \ldots a_1 a_0 \qquad (9)$$

Běžně pro $n > 0$ přidáváme požadavek, aby koeficient a_k u nejvyšší mocniny základu g byl nenulový; pak je vyjádření čísla n v uvedeném tvaru jednoznačné a říkáme, že n je (k+1)-ciferné číslo. K vyjádření čísla 0 požadavek nenulovosti koeficientu u nejvyšší mocniny musíme vynechat; nulu zapisujeme jednoprvkovou posloupností 0.

Uvedeme nyní algoritmus pro určení číslic $c_0, c_1, \ldots, c_k$ kterými je vyjádřeno kladné číslo n v soustavě o základu g. Jednotlivé číslice zápisu získáme opakovaným použitím věty o dělení celých čísel.

Algoritmus 1.

Položme $i = 0$, $q = n$.

L:

```
    r  :=  rem(q, g)
    q  :=  div(q, g)
    a_i := r
    Jestliže q ≠ 0, pak
        i := i + 1
        přejdeme zpět k návěští L
    jinak
        k := i
        výstup a_k a_k-1 … a_1 a_0
    konec větvení jestliže
    konec algoritmu
```

Nerovnost (3) zaručuje, že se algoritmus zastaví.

K zápisu záporných čísel rozšiřujeme počet použitých symbolů o "minus", je-li $n < 0$, zapíšeme výše uvedeným způsobem číslo $|n|$ a před ně přidáme symbol "-". Znamená to však, že k takto tvořenému zápisu celých čísel potřebujeme n+1 symbolů.

V dodatku 1 ukážeme, co by se stalo, kdybychom popsaný algoritmus pro vyjádření kladných čísel v klasické poziční soustavě použili pro čísla záporná, pro něž nerovnost (3) nemusí platit.

Vraťme se však k zápisu celých nezáporných čísel. V soustavě o základu g existuje g jednociferných čísel, $g^2 - g$ dvojciferných čísel, …, $g^k - g^{k-1} = (g-1)\,g^{k-1}$ k-ciferných čísel, … Kromě jednociferných čísel na prvním místě nemůže být nula, tedy zatímco na ostatních místech může být jedna z g různých číslic, na prvním místě jen jedna z $g - 1$ číslic; první číslice tedy je nositelkou menšího množství informace než číslice ostatní. Dvojkový zápis proto začíná vždy číslicí 1, která se tak stává jen informaci neobsahujícím příznakem začátku zápisu čísla[1].

Základ $g = 10$ se promítá do běžného jazyka, číslovky jsou tvořeny opírajíce se o něj. Počítače pracují se základem 2, k čitelné reprezentaci počítačových zápisů se používají základy $8 = 2^3$, popřípadě $16 = 2^4$. V určitých speciálních případech se lze setkat i s jinými základy číselných soustav (např. 12, 20, 60).

Zápis čísel v pozičních soustavách je jedním z největších objevů v dějinách lidstva. K prioritě desítkové soustavy lze mít výhrady, nicméně jsme si na ni zvykli, slouží nám a promítá se nejen do používaných jazyků, nýbrž jejich prostřednictvím i do celé kultury. Zmínili jsme se o drobných "vadách na kráse", které popsaný systém zápisu má: Pro zápis záporných čísel potřebuje přidaný další znak a na první číslicí je kladeno zmíněné omezení. Naznačíme, jak by je bylo možno odstranit (nikoli zároveň obě). V žádném případě nepůjde o snahu měnit náš zaběhaný systém, nýbrž jen o určitou hru přinášející snad pro někoho nový vhled do světa celých čísel. Aby se kvůli obecnosti neztrácely myšlenky, nebudeme už dále uvažovat obecný základ g, nýbrž se zaměříme jen na dvě hodnoty základu, a to na tradiční hodnou 10 a na netradiční a neprávem opomíjenou hodnotu 3.

2. Beznulová desítková soustava

Seznámíme se teď s ***beznulovou* desítkovou soustavou**, v níž pro každé celé kladné číslo existuje jednoznačné vyjádření a na každé pozici může být jedna z deseti číslic. V dalším textu článku budeme pracovat s některými variantami věty o dělení se zbytkem, přičemž v oddílech 2, 4 a 5 se zaměříme jen vždy na jednu konkrétní hodnotu základu g. V tomto oddíle bude $g = 10$.

[1] Pro zápis nezáporného celého čísla v počítači bývá předem stanoven rozsah čísla, např pro unsigned short Int je to interval $\langle 0; 65535\rangle = \langle 0; 2^{16}-1\rangle$; každé číslo z tohoto intervalu se zapisuje pomocí 16členné posloupnosti dvojkových číslic; a zapisují se tedy i počáteční nevýznamné nuly. Skutečnost, že první nenulová číslice binárního zápisu je ve dvojkové soustavě vždy 1, se využívá v zápise čísel v pohyblivé řádové čárce, jímž bychom se ovšem dostali za hranice množiny celých čísel.

Věta 2. *Nechť n je kladné celé číslo. Pak existují celá čísla q a r taková, že*

$$n = q.10 + r \tag{10}$$

a

$$r \in \{1, \dots, 10\} \tag{11}$$

Vztahem (10) *a podmínkou* (11) *jsou čísla q a r určena jednoznačně.*

I zde platí nerovnost (3), která zajišťuje, že algoritmus pro nalezení zápisu čísla na základě upravené věty o dělení se zbytkem po konečném počtu kroků skončí.

Z věty 2 plyne, že každé kladné celé číslo n můžeme jednoznačně vyjádřit (aniž bychom kladli nějakou doplňující podmínku) součtem[2]

$$n = \Sigma_{i=1}^{k} a_{i-1}\, 10^{i-1}, \tag{12}$$

kde $a_i \in \{1, 2, \dots, 10\}$ ($i = 0, \dots, k$) jsou číslice, pomocí nichž číslo n ve tvaru (9) zapíšeme V tomto vyjádření nepoužíváme symbol pro nulu, ovšem potřebujeme symbol pro 10; v souladu s praxí v oblasti IT použijeme znak A. To, že se zde obejdeme bez nuly, snad potěší všechny ty, kdo nule nechtějí dopřát místo mezi přirozenými čísly.

K určení číslic použijeme opakovaně větu 2, což je mírně upravená věta 1 o dělení se zbytkem; úprava spočívá v tom, že u násobků čísla10 bude nebude zbytek nulový, nýbrž rovný číslu 10.

Algoritmus 2.

Položme $i = 0$, $q = n$.

L:

```
    r  :=  rem(q, g)
    q  :=  div(q, g)
    Jestliže r = 0
        r := 10
        q := q – 1
    konec větve jestliže
```

[2] U beznulové desítkové soustavy se nesetkáváme s možností zápisu čísla s nevýznamnými nulami. Oproti vyjádření (1) ve vyjádření (12) posunujeme o 1 sčítací index.

```
 aᵢ := r
Jestliže q > 0
    i := i + 1
    přejdeme zpět k návěští L
jinak
    k := i
    výstup a_k a_(k-1) … a_1 a_0
konec větvení jestliže
konec algoritmu
```

Protože q se po každém cyklu zmenší a při tom je nezáporné, je zajištěno, že se algoritmus zastaví.

Jak vypadá zápis čísel v beznulové desítkové soustavě, demonstruje tabulka 1. Platí: *Ve vyjádření daného čísla v klasické desítkové soustavě se nevyskytuje* 0, *právě když se v jeho vyjádření v beznulové desítkové soustavě nevyskytuje* A*; v tomto případě jsou obě vyjádření stejná*. V beznulové desítkové soustavě je 10 jednociferných čísel, 100 = 10^2 dvojciferných, …, 10^j čísel j-ciferných.

V beznulové dekadické soustavě můžeme vyjádřit ovšem i nulu, která jako kardinální číslo prázdné (tedy konečné) množiny mezi přirozená čísla patří. Jejím vyjádřením je prázdná posloupnost (v praxi ovšem k jejímu zápisu potřebujeme nějaký metaznak, např. ∅ nebo {}). Výše uvedené tvrzení o počtu k-ciferných čísel můžeme rozšířit i pro k = 0; existuje jedno nulciferné číslo, a to 0. Číslo k ve vyjádření (12) vyjadřuje počet číslic[3] daného čísla, při k = 0 je horní mez menší než dolní mez, jde tedy o prázdný součet, jehož hodnota je 0.

K zápisu záporných čísel v beznulové desítkové soustavě bychom jako v klasické soustavě symbol mohli použít symbol -, nicméně soustava jako taková má své kouzlo pro nezáporná celá čísla (včetně prázdné posloupnosti znaků pro nulu). Samozřejmě i zde lze vytvořit algoritmy pro aritmetické operace. Velkým nedostatkem této soustavy ovšem je, že ač jde o soustavu desítkovou, zápis některých čísel zde nekoresponduje s číslovkami v jazyce. A tak její místo zřejmě zůstane ve světě akademických úvah.

[3] Pro počet číslic $k(n)$ každého přirozenéno čísla n v beznulové desítkové soustavě platí

$k(n) = \mathrm{Int}(\log(9n+1))$;

Int značí celou část. Pro porovnání: počet číslic $k'(n)$ nenulového přirozenéno čísla n v klasické desítkové soustavě je

$k'(n) = \mathrm{Int}(\log(10n))$

3. Soustavy se zápornými číslicemi

Věta o dělení se zbytkem je základem modulární aritmetiky. Zbytek vyjadřuje třídu modulo g, do níž číslo n patří. Algoritmus pro zápis čísel v poziční soustavě, uvedený v předchozích částech článku, však byl použitelný jen pro nezáporná čísla. Uvedeme nyní onu stěžejní větu teorie dělitelnosti v poněkud pozměněném tvaru tak, aby na jejím základě bylo možno vytvořit číselnou soustavu umožňující zápis libovolného celého čísla (aniž by bylo třeba kromě číslic užívat nějaký dodatečný znak).

Věta 3. *Nechť n, g a f jsou celá čísla,* $g \geq 3$, $1 \leq f \leq g - 2$. *Pak existují právě jedna taková dvojice celých čísel q, r, že*

$$n = q.g + r \tag{13}$$

a

$$r \in \{-g + f + 1, \ldots, -1, 0, 1, \ldots, f\} \tag{14}$$

-

Přitom platí:

Je-li $n \neq 0$, *pak*

$$|n| > |q| \; ; \tag{15}$$

pro $n = 0$ *je* $q = r = 0$.

Poziční číselná soustava o základu g, v níž existují číslice obou znamének, umožňuje zapisovat všechna celá čísla jen pomocí g číslic (tedy bez symbolu "minus" či nějakého jiného doplňujícího symbolu), přičemž číslice odpovídají možným hodnotám "zbytku" r ve vztahu (14). Zvláštní pozornost zaslouží soustavy "symetrické kolem nuly", tedy takové, u nichž s každou číslicí a je k dispozici i číslice pro $-a$. Protože popisované soustavy obsahují číslici pro nulu, musejí mít lichý počet číslic, a tedy být jejich základem musí být liché číslo.

Tabulka 1

Klasická	Beznulová
0	{}
1	1
2	2
3	3
9	9
10	A
11	11
12	12
19	19
20	1A
21	21
30	2A
40	3A
80	7A
90	8A
99	99
100	9A
101	A1
102	A2
103	A3
104	A4
105	A5
106	A6
107	A7
108	A8
109	A9
110	AA
111	111
119	119

Klasická	Beznulová
120	11A
121	121
190	18A
200	19A
201	1A1
209	1A9
210	1AA
211	211
890	88A
900	89A
901	8A1
902	8A2
910	8AA
911	911
990	98A
991	991
999	999
1000	99A
1001	9A1
1009	9A9
1010	9AA
1011	A11
1090	A8A
1099	A99
1100	A9A
1101	AA1
1109	AA9
1110	AAA
1111	1111

Soustavy s lichým základem bývají zmiňovány jen velice sporadicky. Lze si klást otázku, proč tomu tak je. Sudý základ umožňuje podle poslední číslice určit paritu zapsaného čísla, to je jistě užitečná vlastnost. Na neoblíbenosti lichých číselných základů se může podílet i negativní zabarvení slova "lichý" v jazyce (u polského "nie parzysty" toto zabarvení není, naproti tomu u anglického "odd" je hodně výrazné). Nicméně je pravda, že číslo 3 má v lidských představách mezi lichými čísly poněkud mimořádné postavení (Sv. Trojice, poloha tuhého tělesa v prostoru je určena 3 body). Trojkové soustavě s číslicemi obou znamének se budeme věnovat v následujícím oddíle; poněvadž je symetrická vůči kladným a záporným číslům, nazveme ji ***symetrickou*** **trojkovou soustavou**.

4. Symetrická trojková soustava

Zápis čísla v ***symetrické trojkové soustavě*** se opírá o větu 3, kde g = 3 a f = 1. Odtud pak plyne:

Každé celé číslo $n \neq 0$ lze jednoznačně vyjádřit ve tvaru

$$n = \Sigma_{i=0}^{k} c_i 3^i, \tag{16}$$

kde $c_k \in \{-1, 1\}$, $c_i \in \{-1, 0, 1\}$ pro $i \in \{0, \ldots, k-1\}$.

Pro čísla -1, 0, 1 používejme jako číslice dále symboly po řadě **n**, **o**, **p** (symboly jsou voleny tak, aby vzhled zápisu byl homogenní; jde o písmena následující bezprostředně po sobě v abecedě, přičemž **o** připomíná nulu a **n** a **p** jsou počáteční písmena slov negativní a pozitivní). Vyjádření nenulového čísla n v symetrické trojkové soustavě pak bude

$$c_k c_{k-1} \ldots c_1 c_0 \tag{17}$$

Číslice c_i opět získáme opakovaným použitím věty 3 pro g = 3 a f = 1:

Nechť n je celé číslo. Pak existují celá čísla q a r taková, že

$$n = q.3 + r \tag{18}$$

a

$$r \in \{-1, 0, 1\} \tag{19}$$

Vztahem (18) *a podmínkou* (19) *jsou čísla q a r určena jednoznačně, přičemž pro $n \neq 0$* platí $|n| > |q|$.

Vztah (15) zaručuje, že proces po konečném počtu kroků skončí.

Nulu zapíšeme **o**. Číslo *n* je kladné, resp. záporné, pokud c_k = **n**, resp. c_k = **p**. Číslo zapsané v symetrické trojkové soustavě je liché, resp. sudé, právě když součet je cifer je lichý, resp. sudý. V tabulce 2 jsou příklady zápisu čísel v této soustavě, z tabulky je patrno, že od zápisu daného čísla k číslu opačnému přejdeme záměnou cifer **n** a **p**. V symetrické trojkové soustavě platí, že součin jednociferných čísel je jednociferný (tab. 3), což zjednodušuje konstrukci algoritmu pro násobení čísel v této soustavě. V dodatku 2 si ukážeme, že trojka je ovšem zajímavá i jako základ klasické poziční soustavy.

Příklad. Pomocí opakovaného využití věty 3 převedeme číslo -60 do symetrické trojkové soustavy:

$$-60 = -20.3 + 0$$

$$-20 = -7 \,.\, 3 + 1$$

$$-7 = -2 \,.\, 3 - 1$$

$$-2 = -1 \,.\, 3 + 1$$

$$-1 = \mathbf{0} \,.\, 3 - 1 \qquad (q = 0 \Rightarrow \text{konec výpočtu})$$

Vyjádření čísla -60 v symetrické trojkové soustavě je **npnpo**

Čísla v symetrické trojkové soustavy lze technicky realizovat pomocí trichotomických stavů (zmagnetováno jedním směrem – nezmagnetováno -, zmagnetováno obráceným směrem, resp. záporné napětí - bez napětí – kladné napětí). Vývoj počítačů zvolil cestu binárního kódování. Možná je to škoda. Trojkový zápis čísel rozhodně pozornost zaslouží a my se k němu v závěru ještě vrátíme.

Tabulka 2 je na následující stránce.

Tabulka 3

x	n	o	p
n	p	o	n
o	o	o	o
p	n	o	p

Tabulka 2

desítkově	v symet. trojk. soust.
0	o
1	p
2	pn
3	po
4	pp
5	pnn
6	pno
7	pnp
8	pon
9	poo
10	pop
11	ppn
12	ppo
13	ppp
14	pnnn
15	pnno
16	pnnp
17	pnon
18	pnoo
19	pnop
20	pnpn
21	pnpo
22	pnpp
23	ponn
24	pono
25	ponp
26	poon
27	pooo
28	poop
29	popn
30	popo
64	pnpop
100	ppnop

desítkově	v symet. trojk. soust.
0	n
-1	n
-2	np
-3	no
-4	nn
-5	npp
-6	npo
-7	npn
-8	nop
-9	noo
-10	non
-11	nnp
-12	nno
-13	nnn
-14	nppp
-15	nppo
-16	nppn
-17	npop
-18	npoo
-19	npon
-20	npnp
-21	npno
-22	npnn
-23	nopp
-24	nopo
-25	nopn
-26	noon
-27	nooo
-28	noon
-29	nonp
-30	nono
-64	npnon
-100	nnpon

5. Posunutá desítková soustava

Na závěr uveďme příklad desítkové soustavy používající číslice pro -1 až 8; pro -1 budeme používat symbol N. Nazvěme ji *posunutou desítkovou soustavou*. Zápis čísla, které v klasické desítkové soustavě neobsahuje číslici 9, je v posunuté desítkové soustavě stejný.

Uvedeme nyní jeden příklad převodu z klasické do posunuté desítkové soustavy, využívající opakované použití věty 3. Na příklad se odvoláme i v dodatku 1.

Příklad. Máme převést číslo -894 do posunuté dvojkové soustavy.

-894 = -90 . 10 + 6

-90 = -9 . 10 + 0

-9 = - 1 . 10 + 1

-1 = 0 . 10 - 1

Zápis čísla -894 v posunuté desítkové soustavě je N106

Různé příklady zápisu čísel v posunuté desítkové soustavě jsou v tabulce 4.

6. Dodatek 1: Doplňkový kód

V různých měřidlech (elektroměry, plynoměry, vodoměry, ukazatele ujeté vzdálenosti ve vozidlech aj.) se setkáváme s tím, že zobrazujeme čísla na displeji s určitým omezeným počtem míst, např. na pětimístném displeji můžeme zobrazit kladná celá čísla od 0 (zapsaná 00000) do 99999. Pokud by se takové počítadlo používalo dále, po připočtení 1 hodnotě 99999 by se ukázala hodnota 0. Tato počítadla tak vlastně nepracují s aritmetikou v oboru integrity celých čísel, nýbrž v okruhu zbytkových tříd podle modulu 10^5. Oněch 10^5 zbytkových tříd lze interpretovat různě; můžeme např. chtít zobrazovat čísla obou znamének, která jsou v absolutní hodnotě menší než 50000; čísla zapsaná jako 50001 až 99999 pak budeme interpretovat jako s nimi kongruentní (mod 10^5) záporná čísla z intervalu $\langle -49999; -1 \rangle$.[4] Při této konvenci bude číslo -894 zapsáno jako -894 + 100000 = 99106. Tento zápis záporných čísel nazýváme ***doplňkovým kódem***; používá se běžně v

[4] K interpretaci zápisu ještě doplníme, zda zápis 50 000 bude znamenat kladné číslo 50 000 či záporné číslo -50 000.

počítačích (při použití dvojkové soustavy) k zápisu *celých čísel (se znaménkem i bez znaménka)*.

Algoritmus 1 slouží pro získání zápisu kladných čísel. Podívejme se, jak bude pracovat, bude-li mít na vstupu záporné číslo; použijeme shodně s oddílem 4 opět číslo -894. Základ soustavy g = 10.

```
n = -894.
i = 0, q = -894
L: r :=  rem(-894, 10)    r = 6
   q :=  div(-894, 10)    q = -90 ≠ 0
   a0  =  6
   i  = 1
L: r :=  rem(-90, 10)     r = 0
   q :=  div(-90, 10)     q = -9 ≠ 0
   a1 =  0
   i  = 2
L: r :=  rem(-9, 10)     r = 1
   q :=  div(-9, 10)     q = -1 ≠ 0
   a2 =  1
   i  = 3
L: r :=  rem(-1, 10)     r = 9
   q :=  div(-1, 10)     q = -1 ≠ 0
   a3 =  9
   i  = 4
L: r :=  rem(-1, 10)     r = 9
   q :=  div(-1, 10)     q = -1 ≠ 0
   a4 =  9
   i  =  5
atd.
```

Tabulka 4

0	0
1	1
2	2
8	8
9	1N
10	10
11	11
12	12
18	18
19	2N
20	20
88	88
89	1NN
90	1N0
91	1N1
92	1N2
97	1N7
98	1N8
99	10N
100	100
101	101
102	102
109	11N
110	110
111	111
112	112
119	12N
188	188
189	2NN
190	2N0
191	2N1
192	2N2
198	2N8
199	20N
200	2N0
888	888
889	1NNN
900	1N00

0	0
-1	N
-2	N8
-8	N2
-9	N1
-10	N0
-11	NN
-12	N88
-18	N82
-19	N81
-20	N80
-88	N12
-89	N11
-90	N10
-91	N1N
-92	N08
-97	N03
-98	N02
-99	N01
-100	N00
-101	N0N
-102	NN8
-109	NN1
-110	NN0
-111	NNN
-112	N888
-119	N881
-188	N812
-189	N811
-190	N810
-191	N809
-192	N808
-198	N802
-199	N801
-200	N800
-888	N112
-889	N111
-900	N100

Poslední dva průchody cyklem probíhají se všemi stejnými hodnotami, proces by tak běžel do nekonečna, pro $i \geq 3$ dostaneme $a_i = 9$. Na našem pětimístném displeji se tak zobrazí 99106, tedy číslo -894 zapsané v doplňkovém kódu. Do takovéhoto zacyklení se dostane algoritmus 1, který je konstruován pro kladná celá čísla, když je aplikován na záporné číslo; místo zastavení produkuje nekonečnou posloupnost devítek. Když je vnějším zásahem zastaven (případně omezením počtu cyklů podle zadaného počtu míst displeje) , lze výstup interpretovat jako záporné číslo zapsané v doplňkovém kódu.

7. Dodatek 2: Náročnost kódu

Uvažujme situaci, že máme na displeji s n místy zobrazovat v soustavě o základu g kladná celá čísla menší nebo rovná předem zadanému číslu K. Je zřejmé, že délka displeje je klesající funkcí základu g, n musí být větší než $\log_g K$, a tedy stačí volit

$$n = [\log_g K] + 1$$

Zvolme pro další úvahy určité konkrétní dostatečně velké K, pak $n \approx \log_g K$.

Náročností kódu budeme rozumět součin $n.g$ (tedy počtu číslic v soustavě a délky displeje) a budeme hledat základ g, pro nějž je náročnost nejmenší. Pro jednoduchost položme $n = \log_g K$. Máme tedy minimalizovat funkci

$$\mathrm{A}(g) = g.\log_g K = g.\ln K / \ln g = C.\, g / \ln g,$$

na množině {2, 3, 4, … }, kde $C = \ln K$ je kladná multiplikativní konstanta. Na funkci A se můžeme dívat jako na funkci reálné proměnné definovanou na (0, +∞), její derivace

$$\mathrm{A}'(g) = C\,(\ln g - 1) / (\ln g)^2$$

je rovna 0 pro g = e, pro menší hodnoty argumentu je záporná, pro větší pak kladná, a tak A jako funkce reálné proměnné je klesající na (0; e⟩ a rostoucí na ⟨e, +∞); v e má minimum. Na oboru kladných celých čísel větších nebo rovno 2 můžeme minima nabývat v nejbližších celočíselných bodech, tedy v bodě 2 nebo 3; vypočítáme

$$\mathrm{A}(2) = C.2/\ln 2 = 2{,}89\ C$$

$$\mathrm{A}(3) = C.3/\ln 3 = 2{,}73\ C$$

Nejméně náročným, tj. nejúspornějším kódem je tedy trojková soustava (a to jak klasická, tak symetrická). A tak můžeme číslu 3 jako základu poziční číselné soustavy do budoucna popřát více pozornosti v procesu matematického vzdělávání, a připojit podobné přání celé matematice ve vzdělávacím systému.

RNDr. Jiří Nečas

JorgeMalTiempo@seznam.cz