Srednja šola Črnomelj
Kidričeva 18a
8340 Črnomelj

Ročno korenjenje

Matematika ali logika

Avtor:
Vid Kavčič, 2. aG
Srednje šole Črnomelj

Mentor:
Tilen Šetina,
prof. mat.

Črnomelj, 4. junij 2020

Kazalo

1 Zahvala

Prva zahvala gre moji učiteljici Darinki Rogina iz osnovne šole. Ne le za to, da me je navdušila in podala idejo, na kateri sloni celotna raziskovalna naloga, pač pa tudi zato, ker me je že v 5. razredu navdušila nad matematiko samo. Takšno navdušenje pa – vsaj v neki obliki – ostaja za vedno in je neuničljivo. Brez njega pa te raziskovalne naloge pač ne bi bilo. Hkrati pa je naloga poklon čudovitosti dojemanja matematike in sveta, ki mi ga je učiteljica zapustila.

Takoj na drugem mestu – a zato nič manjša – pa gre zahvala mojemu profesorju matematike Tilnu Šetini, legalnemu mentorju te raziskovalne naloge. Za ves čas, delo, popravke, nasvete, ki jih je prispeval k nalogi, nazadnje pa še za poplačane račune za pice in sokove, ki sva jih zaužila pri rezljanju in piljenju končnega izdelka.

In pa profesorju Miroslavu Plutu, ki je imenitno vskočil in poskrbel za angleško različico povzetka.

2 Povzetek

Na internetu najden postopek za ročno korenjenje najprej poskušamo razumeti in ga uporabiti, nadalje še utemeljiti. Najprej definiramo osnovno matematično teorijo korena in navedemo nekaj njegovih lastnosti. Posvetimo se zapisu števil, kjer poleg običajnega desetiškega sestava na novo uvedemo tako imenovani posplošeni desetiški sestav. Spoznamo funkcijo celi del in matematično indukcijo.

Z namenom utemeljitve algoritma definiramo določene spremenljivke, s katerimi lahko korake v postopku za ročno korenjenje utemeljimo teoretično. Pri utemeljevanju algoritma najprej to storimo za korene kvadratov dvomestnega naravnega števila, potem za korene kvadratov trimestnega naravnega števila. Pred dokazom algoritma za koren poljubnega naravnega števila dokažemo izrek za kvadriranje n-členika. Na vsakem koraku pojasnimo smiselnost posameznih delov postopka. Pojasnimo ga tudi za korenjenje decimalnih števil.

Po utemeljitvi algoritma za kvadratni koren poskušamo iznajti še algoritem za ročno tretje korenjenje, ki ga sicer najdemo, vendar ugotovimo, da v praktičnem smislu nima za človeka uporabne vrednosti, a je vendar izredno zanimiv z matematičnega vidika. V zadnjem delu poskusimo še, kako bi teoretično poiskali algoritem za ročno k-to korenjenje, kjer prav tako navedemo nekaj ugotovitev in predvidevanj, ki jih oblikujemo na podlagi prejšnjih dokazov.

Ključne besede

algoritem za ročno korenjenje, kvadratno korenjenje, kubično korenjenje, k-to korenjenje, korenjenje decimalnih števil, desetiški sestav, posplošeni desetiški sestav

Abstract

Firstly, we try to understand and use the procedure of deviceless rooting that has been found on the Internet, then we try to found it properly. First, we define the basic mathematical theory of the root and name some of its characteristics. We devote our attention to writing down the numbers. Apart from ordinary decimal system we introduce the so called general decimal system. We get out know the floor and ceil function and mathematical induction.

In order to found the algorithm we define certain variables that help us theoretically found the steps of deviceless rooting procedure. While founding the algorithm, we first do it for square roots of squares of two-digits natural numbers, after that for roots of squares of three-digits of natural numbers. Before the proof of the algorithm for a root of a random natural number we prove the theorem for squaring the n-term. At each step we clarify the meaning of each step of the procedure, as well for the decimal numbers rooting.

After founding the algorithm for the square root we try to invent the algorithm for deviceless cube rooting. We can find it but soon we realise it is useless as it has no practical value. It is very interesting from mathematical point of view, though. In the last part, we additionally have try at how to find – theoretically – an algorithm for deviceless k-th-rooting, where we mention some conclusions and assumptions that we form on the basis of previous proofs.

Key words

Algorithm for Deviceless Root Extraction, Square Root, Cube Root, k-th Root, Decimal Number Root, Decimal Number System, General Decimal Number System

3 Uvod

Korenjenje (seveda v matematičnem smislu) pride prvič na vrsto v 8. razredu osnovne šole. Bilo je leta 2016, ko smo se pri dodatnem pouku iz matematike pri učiteljici Darinki Rogina ubadali z zahtevnejšimi problemi s tega področja. Pri eni od nalog smo morali uporabiti računalo, saj je bil korenjenec zelo velik. Učiteljica, ki je tistega leta pravzaprav zaključevala svojo dolgoletno učiteljsko kariero, je po tem, ko smo dobili rešitev, povedala, da so v času njenih osnovnošolskih let, ko računal še ni bilo, tudi večja števila, ki jih danes korenimo s kalkulatorjem, korenili s prav posebnim postopkom. Povprašal sem jo, kakšen je ta postopek, odvrnila pa je, da je že zelo dolgo tega, kar ga je uporabila nazadnje, in da se ga ne spomni.

Koreniti ročno – brez računala! Kako le me ne bi navdušilo, to sem si zares želel znati.

Zato sem "guglal" in preiskal vse slovenske in angleške zadetke, a tega zanimivega postopka nikakor nisem našel. Po dolgotrajnem iskanju sem le izbrskal neko srbsko spletno stran [**1**], forum, na katerem je algoritem razložen na primeru. Takoj sem ga pogledal, naštudiral ter veselo in navdušeno korenil po dolgem in počez.

Minilo je leto, potem pa še eno in znašel sem se v gimnaziji, kjer pri matematiki vprašanja tipa "Kaj?" in "Kako?" preidejo na "Zakaj?". In ravno tako sem se vprašal, zakaj tega čudežnega postopka za ročno korenjenje, ki ni naveden v nobenem učbeniku, ni moč najti nikjer na internetu . . . zakaj sploh deluje?

Zakaj torej?

4 Načrt dela

Algoritem za ročno kvadratno korenenje, najden na spletu, želimo formalno utemeljiti in dokazati, saj dokaza na spletu nismo zasledili. Torej napraviti zaporedje logičnih sklepov, da bi se prepričali o veljavnosti algoritma. Pri tem si lahko pomagamo s svojimi definicijami in izreki, ki jih sami tudi dokažemo. Šele takrat smo lahko namreč zares prepričani, da algoritem velja v vsej splošnosti. Poleg tega pa s tovrstno nalogo dopolnimo prostor slovenske matematične literature, v katerem se o tej tematiki še ni pisalo.

Dokaze, torej sosledja argumentov, želimo izdelati sami, pri tem pa se utemeljevanja lotiti najprej na poenostavljeni primerih, nadalje pa dokazovanje posplošiti.

Ko je algoritem dokazan, postane neizpodbitna resnica, vedno velja, je resničen v vsej splošnosti. Tako bi ga teoretično lahko uporabili tudi za izdelavo programa, neke vrste računala, ki bi izvajal operacijo kvadratni koren po tem postopku, kar sicer morda ne bi bilo tako praktično kot numerične metode, na katerih temeljijo sodobna žepna računala. Glede na to se zavedamo, da s tem ne bi odkrili nič novega (koreniti v današnjem času menda znamo), vseeno pa bi bil naš postopek, utemeljitev oziroma dokaz drugačen in izviren. Novi in izvirni dokazi, četudi za že dokazana dejstva in izreke, pa se v zgodovini matematike kažejo kot pogoste prakse.

5 Teoretične osnove

5.1 Koren

Definicija 5.1 *Naj bo $n \in \mathbb{N}$ in $a \in \mathbb{R}$. Potem imenujemo n-ti koren iz a $\big($oznaka: $\sqrt[n]{a}\big)$:*

a) nenegativno rešitev enačbe $x^n = a$, če je $n = 2k$, $k \in \mathbb{N}$,

b) rešitev enačbe $x^n = a$, če je $n = 2k - 1$, $k \in \mathbb{N}$.

V izrazu $\sqrt[n]{a^m}$ imenujemo n **korenski eksponent**, m **potenčni eksponent**, potenco a^m pa **korenjenec**.

Izrek 5.1 *(lastnosti korena)*

a)

$$\sqrt[n]{a^m} = \sqrt[np]{a^{mp}}$$

b)

$$\sqrt[n]{a} \cdot \sqrt[n]{b} = \sqrt[n]{ab}$$

c)

$$\frac{\sqrt[n]{a}}{\sqrt[n]{b}} = \sqrt[n]{\frac{a}{b}}$$

d)

$$\left(\sqrt[n]{a}\right)^m = \sqrt[n]{a^m}$$

e)

$$\sqrt[m]{\sqrt[n]{a}} = \sqrt[mn]{a}$$

Dokaz: Dokažimo samo pravilo iz točke e), saj ga bomo potrebovali v nadaljevanju.
Naj bo $\sqrt[m]{\sqrt[n]{a}} = x$. Po definiciji m-tega korena je to ekvivalentno

$$\sqrt[n]{a} = x^m.$$

Po isti definiciji, vendar za n-ti koren, sedaj lahko pišemo

$$a = (x^m)^n.$$

Upoštevamo pravilo za računanje s potencami z naravnimi eksponenti, ki pravi, da je $(x^m)^n = x^{mn}$:

$$a = x^{mn}.$$

Po definiciji (mn)-tega korena pa to pomeni, da je

$$\sqrt[mn]{a} = x = \sqrt[m]{\sqrt[n]{a}}.$$

□

5.2 Pisanje števil

Pomembno bo ločiti med števko in številom.

Definicija 5.2 ***Število*** *je abstrakten pojem, s katerim opisujemo matematične količine.*

Definicija 5.3 ***Števka*** *je simbol, s katerim zapisujemo števila.*

Zaradi preglednosti pri razmišljanju o tematiki raziskovalne naloge bomo indekse v definiciji desetiškega sestava nekoliko prilagodili. Morda se bo na prvi pogled to zdelo nepraktično, saj se klasične oznake indeksov ujemajo z eksponenti desetiških potenc. A vendar, namesto običajnega splošnega zapisa n-mestnega števila $\overline{a_{n-1}a_{n-2}\ldots a_1a_0}$ tako definirajmo

Definicija 5.4 *(desetiški sestav) Naj bo $N \in \mathbb{N}$. Zapisu $\overline{a_1a_2\ldots a_{n-1}a_n}$ števila N pravimo zapis v* ***desetiškem številskem sestavu****, če je*

$$N = \overline{a_1a_2\ldots a_{n-1}a_n} = 10^{n-1}\cdot a_1 + 10^{n-2}\cdot a_2 + \cdots + 10^1\cdot a_{n-1} + 10^0\cdot a_n,$$

pri čemer velja $a_i \in \{0,1,2,3,4,5,6,7,8,9\}$ za $i \in \{2,3\ldots n\}$ in $a_1 \in \{1,2,3,4,5,6,7,8,9\}$.

Definicija 5.5 *(posplošeni desetiški sestav) Naj bo število $N \in \mathbb{N}$. Število lahko v* ***posplošenem desetiškem sestavu*** *izrazimo kot*

$$\underline{\overline{a_1a_2\ldots a_{n-1}a_n}} = 10^{n-1}\cdot a_1 + 10^{n-2}\cdot a_2 + \cdots + 10^1\cdot a_{n-1} + 10^0\cdot a_n,$$

kjer velja $a_i \in \mathbb{N}\cup\{0\}$ za $i \in \{1,2\ldots n\}$.

Posplošeni desetiški sestav se od običajnega torej razlikuje v tem, da za števke lahko vzamemo poljubna naravna števila (vključno z 0), ne pa le cela števila od 0 do 9.

Primer. Število $\underline{\overline{1(23)4}}$, ki je zapisano v posplošenem desetiškem zapisu, lahko v običajnem desetiškem sestavu zapišemo kot

$$\underline{\overline{1(23)4}} = 10^2\cdot 1 + 10^1\cdot 23 + 10^0\cdot 4 = 334 = \overline{334}.$$

Zaradi preglednosti moramo večmestne števke ločiti z oklepaji.

Podobno kot pri običajnem desetiškem sestavu, kjer zapis števil ni enoličen (na primer $4,\overline{9} = 5$), enako oziroma še bolj izrazito to velja v posplošenem desetiškem sestavu. Znamenito število 1729 lahko v posplošenem desetiškem sestavu zapišemo na primer kot $1729 = \underline{\overline{(16)(11)(19)}}$, saj je $16\cdot 100+11\cdot 10+19\cdot 1 = 1729$, ali pa na primer kot $\underline{\overline{(15)(13)(99)}}$, saj je $15\cdot 100+13\cdot 10+99\cdot 1 = 1729$. Vsled tega bi se uvedba novega sestava, kjer zapis ni enolično določen, zdela povsem nesmiselna. Toda zapis bomo v nadaljevanju pogosto uporabili, zato ga tukaj definiramo.

5.3 Celi del

Definicija 5.6 *Naj bodo $x \in \mathbb{R}$ in $m \in \mathbb{Z}$.*

a) Spodnji celi del. $\lfloor x \rfloor = \max\big\{m \in \mathbb{Z}; m \leq x\big\}$

b) Zgornji celi del. $\lceil x \rceil = \min\big\{m \in \mathbb{Z}; m \geq x\big\}$

Neposredno iz definicije sledijo nekatere lastnosti funkcij celi del.

Izrek 5.2 *(lastnosti funkcije celi del) Naj bo $x \in \mathbb{R}$ in $m \in \mathbb{Z}$. Potem velja:*

a) $\lfloor x \rfloor = m \iff m \leq x < m + 1;$

b) $\lfloor x \rfloor = m \iff x - 1 < m \leq x;$

c) $\lceil x \rceil = m \iff m - 1 < x \leq m;$

d) $\lceil x \rceil = m \iff x \leq m < x + 1.$

5.4 Matematična indukcija

Za naravna števila velja poseben način sklepanja, **popolna** ali **matematična indukcija**. Anton Cedilnik [**2**, stran 64] zapiše, da če neka lastnost (izjava) L velja za prvo naravno število (za katero je L smiselna lastnost) in če iz dejstva, da velja za neko naravno število, lahko sklepamo, da velja tudi za njegovega naslednika, lastnost L velja za vsa naravna števila.

6 Kvadratni koren

6.1 Predstavitev algoritma za računanje kvadratnega korena

Algoritem smo, kot že omenjeno, našli na spletni strani *Matemanija* [**1**], kjer je bilo ponazorjeno, kako deluje na primeru decimalnega števila 2.754, 1504. Poglejmo si postopek na podobnem, a vseeno nekoliko drugačnem primeru; poskusimo koreniti število 261, 7924. Namesto enačaja, kot je tako v omenjeni literaturi, bomo rabili simbol $\vDash$, saj enačaj v večini primerov ni ustrezen.

Najprej število razdelimo na bloke po dve števki, gledano od decimalne vejice na levo, prav tako od decimalne vejice na desno.

$$\sqrt{2|61|,79|24} =$$

Pri tem se lahko zgodi, da je v prvem bloku na skrajni levi samo ena števka, vendar s tem ni nič narobe. Temu je tako tudi v našem primeru.

Korenjenja se lotimo na bloku skrajno levo, to je 2. Postavimo si vprašanje: *"Katero je največje naravno število, katerega kvadrat je manjši ali enak* 2*?"* To je število 1, saj je $1^2 = 1$, kar pa je manjše od 2.

Desno od znaka za enakost pišemo to število, torej 1, a pod prvim blokom 2 pišemo kvadrat dobljenega števila, torej 1.

$$\begin{array}{l} \sqrt{2|61|,79|24} = 1 \\ \ 1 \end{array}$$

Sedaj od 2 odštejemo 1 in rezultat, 1, prepišemo spodaj. Nadalje prepišemo še števki iz naslednjega bloka, tj 61.

$$\begin{array}{l} \sqrt{2|61|,79|24} = 1 \\ \ \underline{1} \\ \ 1\ 61 \end{array}$$

Za tem dopišemo poseben simbol $\vDash$, desno od njega naš dozdajšnji rezultat, torej 1, pomnožen z 2, kar pa pomeni 2.

$$\begin{array}{l} \sqrt{2|61|,79|24} = 1 \\ \ \underline{1} \\ \ 1\ 61 \vDash 2\_ \cdot \_ \end{array}$$

In si postavimo vprašanje: *"Katera je največja števka, ki je lahko dopisana številu* 2*, da dobljeno število, pomnoženo s to števko, da rezultat, ki je manjši ali enak* 161*? "*

Odgovor je 6, velja namreč $26 \cdot 6 = 156 \leq 161$, za števko 7 pa to ne bi več veljalo: $27 \cdot 7 = 189$, kar pa ni manjše ali enako 161.

Po razmisleku na prazna polja (pripravljene črte) vpišemo števko 6, števko 6 pa vpišemo tudi v rezultat zgoraj, desno od števke 1. Do zdaj smo obdelali vse bloke do decimalne vejice, zato moramo zdaj v rezultat desno od 6 napisati decimalno vejico.

$$
\begin{array}{l}
\sqrt{2|61|,79|24} = 16, \\
\underline{1} \\
1\ 61 \models 2\underline{6} \cdot \underline{6}
\end{array}
$$

Zdaj izvršimo množenje $26 \cdot 6 = 156$ in rezultat zapišemo pod 161; nato ti dve števili odštejemo.

$$
\begin{array}{l}
\sqrt{2|61|,79|24} = 16, \\
\underline{1} \\
1\ 61 \models 2\underline{6} \cdot \underline{6} \\
\underline{1\ 56} \\
\quad 5
\end{array}
$$

Prepišemo števki iz naslednjega bloka, to je 79:

$$
\begin{array}{l}
\sqrt{2|61|,79|24} = 16, \\
\underline{1} \\
1\ 61 \models 2\underline{6} \cdot \underline{6} \\
\underline{1\ 56} \\
\quad 5\ \ 79
\end{array}
$$

Za tem dopišemo simbol $\models$, desno od njega pa dosedanji rezultat, 16, pomnožen z 2, torej 32:

$$
\begin{array}{l}
\sqrt{2|61|,79|24} = 16, \\
\underline{1} \\
1\ 61 \models 2\underline{6} \cdot \underline{6} \\
\underline{1\ 56} \\
\quad 5\ \ 79 \models 32\_ \cdot \_
\end{array}
$$

In si kot malo prej zastavimo vprašanje: *"Katera je največja števka, ki je lahko dopisana številu* 32*, da bo dobljeno število, pomnoženo s to števko, manjše ali enako številu* 579*? "*

Odgovor je 1, ker velja, da je $321 \cdot 1 = 321 \leq 579$, po drugi strani pa naslednja števka, 2, ne bi zadostovala; $322 \cdot 2 = 644$, ker to ne bi bilo manjše ali enako 579.

Sledi, da na prazna polja vpišemo 1, to števko pa pripišemo tudi desno od decimalne vejice zgoraj v rezultatu.

$$
\begin{array}{l}
\sqrt{2|61|,79|24} = 16,1 \\
\underline{1} \\
1\ 61 \models 2\underline{6} \cdot \underline{6} \\
\underline{1\ 56} \\
\quad 5\ \ 79 \models 32\underline{1} \cdot \underline{1}
\end{array}
$$

Sedaj vršimo množenje $321 \cdot 1$ in rezultat, 321, zapišemo pod 579, nato bomo števili odšteli.

$$\begin{array}{l}
\sqrt{2|61|,79|24} = 16,1 \\
\underline{1} \\
1\ 61 \vDash 2\underline{6} \cdot \underline{6} \\
\underline{1\ 56} \\
\quad 5\ \ 79 \vDash 32\underline{1} \cdot \underline{1} \\
\quad \underline{3\ \ 21} \\
\quad 2\ \ 58
\end{array}$$

Pripišemo še zadnji blok števk, torej 24:

$$\begin{array}{l}
\sqrt{2|61|,79|24} = 16,1 \\
\underline{1} \\
1\ 61 \vDash 2\underline{6} \cdot \underline{6} \\
\underline{1\ 56} \\
\quad 5\ \ 79 \vDash 32\underline{1} \cdot \underline{1} \\
\quad \underline{3\ \ 21} \\
\quad 2\ \ 58\ 24
\end{array}$$

Potem dopišemo simbol $\vDash$, desno od njega pa dosedanji rezultat (pri čemer zanemarimo decimalno vejico), pomnožen z 2, torej 322:

$$\begin{array}{l}
\sqrt{2|61|,79|24} = 16,1 \\
\underline{1} \\
1\ 61 \vDash 2\underline{6} \cdot \underline{6} \\
\underline{1\ 56} \\
\quad 5\ \ 79 \vDash 32\underline{1} \cdot \underline{1} \\
\quad \underline{3\ \ 21} \\
\quad 2\ \ 58\ 24 \vDash 322_ \cdot _
\end{array}$$

Vnovič postavimo vprašanje: *"Katera je največja števka, ki je lahko dopisana številu* 322*, da bo dobljeno število, pomnoženo s to števko, manjše ali enako številu* 25824*?"* Odgovor je 8, ker velja, da je $3228 \cdot 8 = 25824 \leq 25824$. V tem primeru velja kar enakost. Sledi, da na prazna polja vpišemo 8, to števko pa pripišemo tudi desno od decimalne vejice zgoraj v rezultatu.

$$\begin{array}{l}
\sqrt{2|61|,79|24} = 16,18 \\
\underline{1} \\
1\ 61 \vDash 2\underline{6} \cdot \underline{6} \\
\underline{1\ 56} \\
\quad 5\ \ 79 \vDash 32\underline{1} \cdot \underline{1} \\
\quad \underline{3\ \ 21} \\
\quad 2\ \ 58\ 24 \vDash 322\underline{8} \cdot \underline{8}
\end{array}$$

Napravimo množenje $3228 \cdot 8$ in rezultat, 25824, prepišemo pod 25824, za tem števili odštejemo, in jasno dobimo rezultat 0.

$$
\begin{array}{l}
\sqrt{2|61|,79|24} = 16,18 \\
\underline{1} \\
1\,61 \vDash 2\underline{6} \cdot \underline{6} \\
\underline{1\,56} \\
\quad 5\;\;79 \vDash 32\underline{1} \cdot \underline{1} \\
\quad \underline{3\;\;21} \\
\quad 2\;\;58\,24 \vDash 322\underline{8} \cdot \underline{8} \\
\quad \underline{2\;\;58\,24} \\
\qquad\qquad 0
\end{array}
$$

Ko smo dobili ničlo, je algoritem končan. Iskani rezultat je $16,18$ (ki je čisto slučajno z 10 množen približek nekega naključno izbranega števila). Z žepnim računalom preverimo; in res,

$$16,18^2 = 216,7924.$$

6.2 Teoretičen opis algoritma

Algoritem poskušamo teoretično opisati.

1. Korenjenec razdelimo na bloke po dve števki, gledamo od decimalne vejice na levo, prav tako od decimalne vejice na desno. Če je decimalnih mest liho mnogo, na konec števila dodamo števko 0; ta vrednosti števila ne spremeni. V zapisu $\overline{A_1 d_1 d_2 \ldots d_{2n-2}}$ oznaka A_1 predstavlja prvi blok, d_i so preostale števke korenjenca.

2. Začnemo pri bloku skrajno levo. To je torej edini blok, ki ima morda le eno števko. Označimo število, ki se nahaja v njem, z A_1. Poiščemo največje tako števko x_1, da je $x_1^2 \leq A_1$. S tem smo dobili prvo števko rezultata: $x_1 = R_1$. Pod blok A_1 podpišemo razliko $A_1 - x_1^2$, ki jo imenujmo ostanek O_1.

3. K ostanku O_1 pripišemo naslednji blok, to je blok A_2, dobljeno število pa označimo z B_2. Število B_2 je torej $B_2 = 100 \cdot O_1 + A_2 = \overline{O_1 0 A_2}$. Poiščemo največjo tako števko x_2, da je $B_2 \geq \overline{(2 \cdot x_1)\, x_2} \cdot x_2 = C_2$. Na tem mestu med B_2 in C_2 uporabimo poseben simbol $\vDash$, enačaj namreč v večini primerov ne bi bil ustrezen. Dobimo drugo števko rezultata, ki je tako $R_2 = \overline{x_1 x_2}$. Pod B_2 podpišemo $B_2 - C_2$, ki je novi ostanek O_2.

4. Sedaj k ostanku O_2 pripišemo blok A_3, s čimer dobimo število $B_3 = 100 \cdot O_2 + A_3 = \overline{O_2 0 A_3}$. Poiščemo največjo tako števko x_3, da je $B_3 \geq \overline{(2 \cdot \overline{x_1 x_2})\, x_3} \cdot x_3 = C_3$. S tem smo dobili tretjo števko rezultata: $R_3 = \overline{x_1 x_2 x_3}$. Pod B_3 podpišemo $B_3 - C_3 = O_3$, torej novi ostanek.

5. Postopek, zapisan pri zadnjih dveh točkah zgoraj, ponavljamo, dokler ne pridemo do konca števila. Pri morebitni decimalni vejici to prepišemo na ustrezno mesto − torej med tisti števki x_i in x_{i+1} iz rezultata, od katerih smo prvo dobili na podlagi zadnjega bloka pred, drugo pa na podlagi prvega bloka za decimalno vejico korenjenca. Če je pri zadnjem koraku algoritma ostanek $O_n = 0$, smo dobili natančno vrednost korena, sicer le približek.

V raziskovalni nalogi bomo nekatere dele algoritma posebej imenovali, da nam jih ne bo treba vsakič znova definirati. Zato jih definiramo tukaj.

1. A_i (*i*-**ti blok**) je število v i-tem bloku, kjer je prvi blok A_1 tisti najbolj levo.

2. R_i (*i*-**ti rezultat**) je število, ki se na koncu i-tega koraka nahaja v prvi vrsti za enačajem, torej $R_i = \overline{x_1 x_2 \ldots x_i}$.

3. O_i (*i*-**ti ostanek**) je število, definirano kot $O_i = B_i - C_i$ za $i \geq 2$. Posebej definiramo: $O_0 = 0$ in $O_1 = A_1 - x_1^2$.

4. B_i (*i*-**ti zahtevek**) je število, ki ga dobimo, ko k zadnjemu ostanku O_{i-1} pripišemo i-ti blok, torej $B_i = \overline{O_{i-1} 0 A_i}$. Prvi zahtevek je kar enak prvemu bloku; $B_1 = \overline{O_0 0 A_i} = \overline{00 A_i} = A_1$.

5. C_i (*i*-**ta vrednost**) je število, enako izrazu $\overline{(2 \cdot \overline{x_1 x_2 \ldots x_{i-1}})\, x_i} \cdot x_i = \overline{(2 \cdot R_{i-1}) x_i} \cdot x_i$ za $i > 1$. Posebej definiramo prvo vrednost, ta naj bo $C_1 = x_1^2 = R_1^2$.

6.3 Utemeljitev algoritma

6.3.1 Koren kvadrata dvomestnega naravnega števila

Za začetek se ustavimo na korenjenju kvadratov dvomestnih naravnih števil.

Vzemimo na primer znani kvadrat 625 in ga poskusimo koreniti ročno. Razdelimo na bloke po dve števki, tu bo $A_1 = 6$ ostala sama v bloku.

$$\sqrt{6|25} =$$

Največje tako število, katerega kvadrat je manjši od 6, je $x_1 = 2$, kar pišemo za enačaj, dobimo rezultat $R_1 = 2$.

$$\sqrt{6|25} = 2$$

Kvadrat rezultata $R_1^2 = 2^2 = 4$ napišemo pod 6, izračunamo $O_1 = 2$ in pripišemo naslednji blok $A_2 = 25$.

$$\begin{array}{l} \sqrt{6|25} = 2 \\ \underline{4} \\ 2\ 25 \models 4\_ \cdot \_ \end{array}$$

"Katera je največja števka, ki je lahko dopisana številu 4*, da bo dobljeno število, pomnoženo s to števko, manjše ali enako številu* 225*?"* To je 5, ker velja, da je $C_1 = 45 \cdot 5 = 225 \leq 225 = B_1$. Števko $x_2 = 5$ napišemo na črte in zgoraj v rezultat, ki je sedaj $R_2 = 25$. Vrednost $C_1 = 225$ prepišemo pod zahtevek $B_1 = 225$ in ju odštejemo. Ker je ostanek $O_2 = 0$, smo končali.

$$\begin{array}{l} \sqrt{6|25} = 25 \\ \underline{4} \\ 2\ 25 \models 4\underline{5} \cdot \underline{5} \\ \underline{2\ 25} \\ \quad\ 0 \end{array}$$

Seveda smo prav dobili 25, to smo morali na pamet znati že v osnovni šoli . . .

***

Algoritem ponuja veliko ugank, češ zakaj pa gre ravno tako. Poskusimo zato nastaviti enačbo, s katero bi poskušali število koreniti, ta pa nam bi pomagala pri razlagi algoritma na tem enostavnem primeru.

Ker je $10^2 \leq 625 < 100^2$, to pomeni, da bo rezultat korenjenja gotovo dvomestno število, saj kvadrat omejujeta kvadrat najmanjšega in kvadrat največjega dvomestnega naravnega števila. Naj bo to dvomestno število $\overline{x_1x_2}$. Nastavimo torej enačbo in jo razpišimo.

$$\begin{aligned}(\overline{x_1x_2})^2 &= 625\\ (10x_1 + x_2)^2 &= 625\\ 100x_1^2 + 20x_1x_2 + x_2^2 &= 625\end{aligned}$$

V algoritmu najprej določimo x_1. Zakaj in kako to, da je ta enolično določen? Jasno je, da v enačbi ne more biti $x_1 > 2$, saj je že pri $x_1 = 3$ izraz na desni večji od 625. Števka x_1 je torej enaka 1 ali 2. Algoritem veleva, da izberemo največjo takšno števko, katere kvadrat je manjši od 6, v tem primeru je to 2.

Na tem mestu poskusimo do konca, zgornje kočljive točke se bomo dotaknili v nadaljevanju. Naj bo tako $x_1 = 2$. Potlej imamo:

$$\begin{aligned}100 \cdot 2^2 + 20 \cdot 2 \cdot x_2 + x_2^2 &= 625\\ 400 + 40 \cdot x_2 + x_2^2 &= 625\\ 40 \cdot x_2 + x_2^2 &= 225\\ x_2 \cdot (40 + x_2) &= 225.\end{aligned}$$

Vidimo, da ustreza točno tak $x_2 = 5$, saj je $5 \cdot 45 = 225$, kar je popolno. Rezultat korenjenja je torej $\overline{x_1x_2} = 25$.

***

Poskusili smo s korenjenjem konkretnega kvadrata dvomestnega naravnega števila. Poskusimo zdaj posplošti kvadratno korenjenje kvadratov dvomestnega števila.

Razpišimo kvadrat dvomestnega naravnega števila $\overline{x_1x_2}$.

$$\begin{aligned}(\overline{x_1x_2})^2 = (10x_1 + x_2)^2 &= 100x_1^2 + 20x_1x_2 + x_2^2\\ &= 100x_1^2 + x_2 \cdot (20x_1 + x_2)\\ &= 100x_1^2 + x_2 \cdot \underline{\overline{(2x_1)\, x_2}}\end{aligned}$$

Izpeljava zgoraj razjasni algoritem v tem primeru. Ta tukaj sestoji iz dveh faz; v prvi fazi je treba določiti enolično določljivo števko x_1, nato poiskati ostanek, ki ga moramo z drugim členom še zajeti. Sledi iskanje ustreznega x_2. Zgornja izpeljava pravzaprav utemeljuje vprašanje, ki si ga pri algoritmu vselej zastavljamo, oziroma postopanje iz algoritma; torej množenje števke z 2 in ostalo.

Pregledno komentirajmo korake postopka za ročno kvadratno korenjenje kvadrata dvomestnega naravnega števila. Poskusimo biti kar najbolj splošni.

- **Razdelitev na bloka po dva**

  Korenjenec razdelimo na bloka po dve števki. To v vsakem primeru lahko storimo: če je korenjenec štirimestno število, imamo dva bloka po dve števki, če pa je korenjenec trimesten, lahko levo od prve števke pripišemo še eno "števko" 0 in postopamo kot v primeru štirimestnega korenjenca, torej $\overline{A_1 d_1 d_2}$.

  Blok A_1 pravzaprav snuje vrednost $100 \cdot \left\lfloor \frac{A_1}{100} \right\rfloor$. To vrednost pa v največji možni meri zavzamemo z izrazom $100 \cdot x_1^2$. Razliko do prave vrednosti $100 \cdot O_1$, ki je ne zavzamemo, in pridatek $\overline{d_1 d_2}$ pa zavzamemo s številom $\overline{(2x_1)\, x_2} \cdot x_2$. Izraz $100 x_1^2$ je torej neodvisen od $\overline{d_1 d_2}$, ravno zato je edino smiselna razdelitev na bloke po dve števki.

- **Enolična določenost prve števke**

  Algoritem veleva, da število v prvem bloku z leve korenimo in v rezultat pišemo spodnji celi del tega korena. Vprašanje je, zakaj nobeno manjše oziroma večje celo število ne pride v poštev.

  Imejmo število $\overline{A_1 d_1 d_2}$, kjer sta d_1 in d_2 števki. Oznaka A_1 pa je v splošnem števka posplošenega desetiškega sestava, ki zastopa prvi blok, to pa je bodisi enomestno bodisi dvomestno naravno število.

  Algoritem pravi, da velja

  $$x_1 = \left\lfloor \sqrt{A_1} \right\rfloor,$$

  kjer z oglatima uklepajem in zaklepajem označimo spodnji celi del korena. Po točki b) izreka 4.2 to pomeni

  $$\sqrt{A_1} - 1 < x_1 \leq \sqrt{A_1}$$

  oziroma

  $$\left(\sqrt{A_1} - 1\right)^2 < x_1^2 \leq A_1.$$

  S protislovjem dokažimo, da je tak x_1 edini možen. Ločimo dve možnosti:

  1. Namesto x_1, za katerega velja zgornji pogoj, vzemimo $x_1 - 1$.

     Če dokažemo, da $x_1 - 1$ ni ustrezen, potem tudi nobena manjša vrednost za x_1 ne bo. Preverimo, ali lahko s kvadriranjem tudi zdaj nekako dosežemo prvotni kvadrat. V mejnem primeru za drugo števko števila, ki ga kvadriramo, določimo, da je največja možna, torej 9.

     $$\overline{(x_1 - 1)\, 9} = \overline{x_1 0} - 1$$
     $$\left(\overline{(x_1 - 1) 9}\right)^2 = \left(\overline{x_1 0} - 1\right)^2 < \overline{x_1^2 00} \leq \overline{A_1 00} \leq \overline{A_1 d_1 d_0}$$

     Tak izbor za prvo števko rezultata očitno ne ustreza.

  2. Namesto x_1, za katerega velja zgornji pogoj, vzemimo $x_1 + 1$.

     Če dokažemo, da $x_1 + 1$ ni ustrezen, potem tudi nobena večja vrednost za x_1 ne bo. Zdaj preverimo, ali lahko s kvadriranjem tudi zdaj nekako še ujamemo prvotni kvadrat. V mejnem primeru za ostalo števko števila, ki ga kvadriramo, določimo, da je najmanjša možna, torej 0.

$$\left(\overline{(x_1+1)\,0}\right)^2 = 100\,(x_1+1)^2 > 100\,(A_1+1) - 1 = 100A_1 + 99 = \overline{A_1 99} \geq \overline{A_1 d_1 d_2}$$

Prvo neenakost v zgornji vrstici utemeljimo na podlagi sledečega dejstva: Ker je A_1 celo število, velja očitna zveza $\left\lfloor\sqrt{A_1}\right\rfloor = \left\lfloor\sqrt{A_1 + 99/100}\right\rfloor$. Vemo že, da je

$$\sqrt{A_1} - 1 < x_1 \leq \sqrt{A_1}, \text{ torej } \sqrt{A_1} - 1 < x_1,$$

od koder sledi

$$\sqrt{A_1} < x_1 + 1$$

in od tod

$$A_1 < (x_1+1)^2\,.$$

Na podlagi zadnjih ugotovitev lahko zapišemo:

$$\begin{aligned}
& x_1 = \left\lfloor\sqrt{A_1 + 99/100}\right\rfloor \\
\Longrightarrow\quad & \sqrt{A_1 + 99/100} - 1 < x_1 \leq \sqrt{A_1 + 99/100} \\
\Longrightarrow\quad & A_1 + 99/100 < (x_1+1)^2 \\
\Longrightarrow\quad & 100\,(x_1+1)^2 > 100A_1 + 99 \\
\Longrightarrow\quad & 100\,(x_1+1)^2 > 100\,(A_1+1) - 1.
\end{aligned}$$

Tudi tak izbor za prvo števko rezultata očitno ne ustreza.

- **Ostanek**

  Ko določimo x_1, v algoritmu x_1 kvadriramo in ga podpišemo pod prvi blok. Odštejemo pa dobimo ostanek O_1. V resnici pa, kar razjasni razpisan kvadrat dvomestnega števila, od prvotnega števila odštejemo $100x_1^2$. Naslednja števka v rezultatu mora zajeti vrednost tega ostanka.

  Ker korenimo kvadrat dvomestnega naravnega števila, je drugi ostanek očitno $O_2 = 0$.

- **Množenje z** 2

  V algoritmu vsakič, ko določamo drugo števko, dosedanji rezultat $R_1 = x_1$ pomnožimo z 2 in nadalje sledimo korakom algoritma. To množenje pojasni faktor 2 v preoblikovanem izrazu za kvadrat dvomestnega števila:

  $$\left(\overline{x_1 x_2}\right)^2 = 100x_1^2 + x_2 \cdot \overline{(\mathbf{2}x_1)\,x_2},$$

  saj je število x_1 prva števka, ki pa smo jo že določili in vpisali za enačaj zgoraj.

- **Dopisana števka** x_2

  V algoritmu se vsakič, ko določamo drugo števko, vprašamo: *"Katera je največja števka, ki je lahko dopisana dvakratniku dosedanjega rezultata, da dobljeno število, pomnoženo s to števko, da vrednost, ki je manjša ali enaka od zahtevka?"*

  Ker je v preoblikovanem izrazu

$$(\overline{x_1 x_2})^2 = 100x_1^2 + \mathbf{x_2} \cdot \underline{\overline{(2x_1)\mathbf{x_2}}}$$

x_1 dosedanji rezultat in x_2 tista števka, ki jo skušamo dopisati, je ravno ta oblika izraza utemeljitev postopka.

Tako bi si vprašanje lahko zastavili tudi kot: *"Katera števka x_2 naj bo dopisana številu $2x_1$, da lahko dobljeno število, pomnoženo s to števko x_2, da rezultat, ki je manjši ali enak zahtevku?"* Ta je v tem primeru enak $(\overline{x_1 x_2})^2 - 100x_1^2 = x_2 \cdot \underline{\overline{(2x_1)x_2}} = C_2$.

Ker obravnavamo koren kvadrata dvomestnega števila, vemo, da se bo korenjenje izšlo. Tako lahko rečemo, da iščemo tako števko x_2, da je vrednost C_2 enaka zahtevku B_2.

Razmislek, zakaj je x_2 vedno največja taka števka, da ustreza pogoju zahtevka, je preprost. Ker je števka x_1 enolično določena in ker vemo, da korenimo kvadrat dvomestnega naravnega števila, je namreč enolično določena tudi x_2. Ker pri tem vršimo zadnji korak, vsekakor velja, da je zahtevek enak vrednosti oziroma $B_2 = C_2$. Iz tega sledi, da je x_2 kar največji možen, da zadosti neenakosti $B_2 \geq C_2$ – v tem primeru torej kar enakosti $B_2 = C_2$.

6.3.2 Koren kvadrata trimestnega števila

Število 189225 je kvadrat trimestnega naravnega števila (delamo se, da ne vemo, katero število smo kvadrirali v kalkulatorju). Poskusimo ga koreniti z našim algoritmom.

1. Število razdelimo na tri bloke. Vprašamo se: *“Katero je največje naravno število, za katerega velja, da je njegov kvadrat manjši ali enak* 18*?”* To je jasno število 4, saj je $4^2 \leq 18$. Desno od enačaja pišemo prvi rezultat $R_1 = 4$, pod prvi blok $A_1 = 18$ pa $R_1^2 = 4^2 = 16$. Odštejemo, pod črto zapišemo prvi ostanek, se pravi $O_1 = 18 - 4^2 = 2$. Pripišemo naslednji blok A_2.

$$
\begin{array}{l}
\sqrt{18|92|25} = 4 \\
\underline{16} \\
\;\;2\;92
\end{array}
$$

2. Za simbol $\vDash$ pripišemo dvakratnik dosedanjega (prvega) rezultata, torej $2R_1 = 8$, in se vprašamo: *“Katera je največja števka, ki je lahko dopisana številu* 8*, da bo dobljeno število, pomnoženo s to števko, manjše ali enako drugemu zahtevku številu* $B_2 = 292$*?”*

   Z nekaj računanja ugotovimo, da je to število 3, saj je $C_2 = 83 \cdot 3 = 249 \leq 292 = B_2$. Števko 3 napišemo na črti in zgoraj za enačaj in dobimo drugi rezultat $R_2 = 43$. Obravnavani produkt prepišemo pod drugi zahtevek $B_2 = 292$, odštejemo, in pod črto zapišemo drugi ostanek $O_2 = 43$. Pripišemo naslednji blok $A_3 = 25$ in dobimo tretji zahtevek $B_3 = 4325$.

$$
\begin{array}{l}
\sqrt{18|92|25} = 43 \\
\underline{16} \\
\;\;2\;92 \vDash 8\underline{3} \cdot \underline{3} \\
\;\;\underline{2\;49} \\
\;\;\;\;43\;25
\end{array}
$$

3. Spodaj za $\vDash$ pripišemo dvakratnik dosedanjega rezultata $2R_2 = 86$ in se vprašamo: *“Katera je največja števka, ki je lahko dopisana številu* 86*, da bo dobljeno število, pomnoženo s to števko, manjše ali enako zahtevku* $B_3 = 4325$*?”*

   To število 5, saj je $C_3 = 865 \cdot 5 = 4325 \leq B_3 = 4325$. Števko 5 napišemo na črti in v rezultat, da dobimo R_3. Produkt C_3 prepišemo pod $B_3 = 4325$, odštejemo, in ostanek $O_3 = 0$ zapišemo pod črto. Ker je ostanek 0, smo zaključili s korenjenjem. Končni rezultat je torej $R_3 = 435$.

$$
\begin{array}{l}
\sqrt{18|92|25} = 435 \\
\underline{16} \\
\;\;2\;92 \vDash 8\underline{3} \cdot \underline{3} \\
\;\;\underline{2\;49} \\
\;\;\;\;43\;25 \vDash 86\underline{5} \cdot \underline{5} \\
\;\;\;\;\underline{43\;25} \\
\;\;\;\;\;\;\;\;\;\;0
\end{array}
$$

***

Predstavljajmo si zdaj v splošnem kvadrat trimestnega naravnega števila $\overline{x_1x_2x_3}$, za katerega želimo utemeljiti algoritem za ročno korenjenje. To pomeni, da moramo podobno kot pri korenjunju kvadrata dvomestnega števila naš izraz zapisati tako, da bo jasno utemeljil postopanje v algoritmu.

$$\begin{aligned}
(\overline{x_1x_2x_3})^2 &= (100x_1 + 10x_2 + x_3)^2 = 10000x_1^2 + 100x_2^2 + x_3^2 + 2000x_1x_2 + 200x_1x_3 + 20x_2x_3 \\
&= 10000x_1^2 + 2000x_1x_2 + 100x_2^2 + 200x_1x_3 + 20x_2x_3 + x_3^2 \\
&= 10000x_1^2 + 100x_2 \cdot (20x_1 + x_2) + x_3 \cdot (200x_1 + 20x_2 + x_3) \\
&= 10000x_1^2 + 100 \cdot x_2 \cdot \overline{(2x_1)\,x_2} + x_3 \cdot \overline{(2 \cdot (\overline{x_1x_2}))\,x_3}
\end{aligned}$$

Oblika izraza kvadrata trimestnega naravnega števila, ki se nahaja v zadnji vrstici zgoraj, utemeljuje korake algoritma pri korenjenju kvadrata trimestnega naravnega števila.

Kot pri korenu kvadrata dvomestnega števila tudi tokrat nekoliko komentirajmo posamezne korake algoritma.

- **Razdelitev na bloka po dva**

  Korenjenec razdelimo na bloke po dve števki. To v vsakem primeru lahko storimo: če je korenjenec šestmestno število, imamo tri bloke po dve števki, če pa je korenjenec petmesten, lahko levo od prve števke pripišemo še eno "števko" 0 in postopamo kot v primeru šestmestnega korenjenca, torej $\overline{A_1d_1d_2d_3d_4}$.

  Blok A_1 pravzaprav snuje vrednost $10000 \cdot \left\lfloor \frac{A_1}{10000} \right\rfloor$. To vrednost pa v največji možni meri zavzamemo z izrazom $10000x_1^2$. Razlika do prave vrednosti $10000 \cdot O_1$, ki je ne zavzamemo, in pridatek $100 \cdot \overline{d_1d_2}$ pa zavzamemo s številom $100 \cdot \overline{(2x_1)x_2} \cdot x_2$. Novo razliko do prave vrednosti $100 \cdot O_2$, ki je ne zavzamemo z vrednostjo C_2, in pridatek $\overline{d_3d_4}$ pa zavzamemo s številom $x_3 \cdot \overline{(2 \cdot \overline{x_1x_2})\,x_3}$.

  Izraz $10000x_1^2$ je neodvisen od $100 \cdot x_2 \cdot \overline{d_1d_2}$, prav tako je izraz $100 \cdot x_2 \cdot \overline{d_1d_2}$ neodvisen od $x_3 \cdot \overline{(2 \cdot \overline{x_1x_2})\,x_3}$. Zato je tudi tokrat razdelitev v bloke po dve števki smiselna.

- **Prva števka x_1**

  Utemeljitev, da je prva števka zares enolično določena, je v tej točki skorajda povsem skladna tisti pri korenjenju kvadrata dvomestnega naravnega števila. Zaradi tega in tudi dejstva, da splošnejša obrazložitev sledi v naslednjem podpodrazdelku, utemeljitev v tej točki prepustimo bralcu.

- **Prvi ostanek O_1**

  Utemeljitev za prvi ostanek je skorajda enaka tisti pri korenu kvadrata dvomestnega naravnega števila, zato jo prav tako prepuščamo bralcu.

- **Množenje z 2 in dopisani novi števki x_2 in x_3**

  Ostanku O_1 pripišemo blok A_2, tako dobljeni zahtevek $B_2 = \overline{O_1 0 A_2}$ izrazimo z izrazom $100 \cdot x_2 \cdot \overline{(2x_1)x_2}$, ki pa je pravzaprav identičen navodilu oziroma vprašanju: *"Katera je največja števka x_2, ki je lahko dopisana številu $2 \cdot x_1$, da bo dobljeno število, pomnoženo s to števko, manjše ali enako zahtevku?"* Faktor 100 je smiseln, saj blok A_2 predstavlja stotice.

Z izrazoma $10000x_1^2$ in $100 \cdot x_2 \cdot \overline{(2x_1)x_2}$ smo do zdaj zajeli del kvadrata, ki ga korenimo. Novi razliki do prave vrednosti korenjenca, torej zahtevku B_2, se približamo z izrazom $x_3 \cdot \overline{(2 \cdot (\overline{x_1 x_2}))\, x_3} = C_3$, ki pa je ekvivalenten vprašanju: *"Katera je največja števka x_3, ki je lahko dopisana številu $2 \cdot \overline{x_1 x_2}$, da bo dobljeno število, pomnoženo s to števko, manjše ali enako številu, ki ga predstavlja tretji zahtevek?"*

Ker obravnavamo koren kvadrata trimestnega števila, vemo, da se bo korenjenje izšlo. Tako lahko rečemo, da iščemo tako števko x_3, da je vrednost C_3 enaka zahtevku B_3.

Razmislek, zakaj sta x_2 in x_3 vedno največji taki, da ustrezata pogoju zahtevka, prepuščamo bralcu. Lastnost bomo v splošnem utemeljili v naslednjem podpodrazdelku.

- **Drugi in tretji ostanek O_2 in O_3**

  Ko določimo x_2, drugi ostanek določimo kot $O_2 = B_2 - C_2$. V skladu s prejšnjo točko nato določimo števko x_3. Ker korenimo kvadrat trimestnega naravnega števila, jasno velja $O_3 = B_3 - C_3 = 0$.

6.3.3 Posplošitev na koren kvadrata n-mestnega števila

Algoritem poskušamo utemeljiti v vsej splošnosti; torej za kvadratno korenjenje kvadrata n-mestnega števila.

***

Zanimiva je ocena, koliko mest ima koren nekega popolnega kvadrata. Izkaže se sicer, da tega pri sami utemeljitvi algoritma za ročno korenjenje ne potrebujemo, vendar je ugotovitev sama po sebi zanimiva, zato jo vključimo v nalogo.

Zamislimo si neko n-mestno naravno število N. Potem velja

$$10^{n-1} \leq N < 10^n.$$

Če to neenakost kvadriramo, dobimo

$$\begin{aligned} & 10^{2n-2} \leq N^2 < 10^{2n} \\ \iff \quad & 10^{2n-2} \leq N^2 \leq 10^{2n} - 1 \\ \iff \quad & 1\underbrace{00\ldots0}_{(2n-2)\text{-krat}} \leq N^2 \leq \underbrace{99\ldots9}_{(2n)\text{-krat}} \end{aligned}$$

Poudarimo, da ima levo število v zadnji vrstici $2n-1$ števk, od tega $2n-2$ ničel in eno enico.

Zgornje ugotovitve lahko povemo drugače: Če je število števk korenjenca $\big($tu označen z $N^2\big)$ enako $2n-1$ ali $2n$, ima njegov koren (zgoraj zapisan kot število N) ravno $n = \left\lceil \frac{2n-1}{2} \right\rceil = \left\lceil \frac{2n}{2} \right\rceil$ števk.

To se ravno ujema s številom blokov, na katere razdelimo začetno število: če ima korenjenec sodo mnogo mest, torej $2n$, je blokov $\frac{2n}{2} = n$, če pa ima število pod korenom liho mnogo mest, torej $2n-1$, spredaj dodamo eno števko 0, s tem dobimo $2n$ mest in n blokov. In vsak blok, kot vemo, doprinese k eni števki v rezultatu, ki je n-mestno število.

Ugotovitev iz tega razdelka lahko zapišemo še splošno: če ima korenjenec, popoln kvadrat nekega naravnega števila, m mest, ima njegov koren $\left\lceil \frac{m}{2} \right\rceil$ mest.

***

Pri obeh prejšnjih utemeljitvah algoritma smo najprej kvadrirali dvočlenik oziroma tričlenik, da smo dobili kvadrat števila, ki smo ga nadalje preučevali. Tako bomo morali tudi v vsej splošnosti kvadrirati – zdaj n-členik. To pa ni tako enostavno kot kvadriranje dvočlenika in tričlenika. Vsled tega je pametno, da uganemo in dokažemo izrek za kvadriranje n-členika.

Izrek 6.1 *(kvadrat n-členika)*

$$(x_1 + x_2 + \cdots + x_n)^2 = \sum_{\substack{i,j=1\\ i<j}}^{n} 2x_i x_j + \sum_{i=1}^{n} x_i^2 = \sum_{i,j=1}^{n} x_i x_j \tag{6.3.1}$$

Dokaz: Dokažimo s popolno ali matematično indukcijo.

Baza indukcije: Pri $n = 1$ trditev drži, saj

$$x_1^2 = \sum_{\substack{i,j=1\\ i<j}}^{1} 2x_i x_j + \sum_{i=1}^{1} x_i^2 = 0 + \sum_{i=1}^{1} x_i^2 = x_1^2$$

Indukcijski korak: Privzemimo indukcijsko predpostavko, da trditev že velja za nek $n \in \mathbb{N}$:

$$(x_1 + x_2 + \cdots + x_n)^2 = \sum_{\substack{i,j=1\\ i<j}}^{n} 2x_i x_j + \sum_{i=1}^{n} x_i^2.$$

Dokažimo, da velja tudi za $n + 1$:

$$(x_1 + x_2 + \cdots + x_n + x_{n+1})^2 = \sum_{\substack{i,j=1\\ i<j}}^{n+1} 2x_i x_j + \sum_{i=1}^{n+1} x_i^2.$$

- -

Preoblikujemo levo stran v desno.

$$\begin{aligned}
&(x_1 + x_2 + \cdots + x_n + x_{n+1})^2 = \\
&= ((x_1 + x_2 + \cdots + x_n) + x_{n+1})^2 = \\
&= (x_1 + x_2 + \cdots + x_n)^2 + 2x_{n+1}(x_1 + x_2 + \cdots + x_n) + x_{n+1}^2 = \\
&= \sum_{\substack{i,j=1\\ i<j}}^{n} 2x_i x_j + \sum_{i=1}^{n} x_i^2 + 2x_{n+1} \sum_{i=1}^{n} x_i + x_{n+1}^2 = \\
&= \left(\sum_{\substack{i,j=1\\ i<j}}^{n} 2x_i x_j + 2x_{n+1} \sum_{i=1}^{n} x_i \right) + \left(\sum_{i=1}^{n} x_i^2 + x_{n+1}^2 \right) = \\
&= \sum_{\substack{i,j=1\\ i<j}}^{n+1} 2x_i x_j + \sum_{i=1}^{n+1} x_i^2
\end{aligned}$$

$\square$

Pravkar dokazani izrek lahko zdaj uporabimo pri dokazovanju našega ciljnega izreka − algoritma za ročno korenjenje.

Izrek 6.2 *(za algoritem za ročno korenjenje)*

$$
\begin{aligned}
&(\overline{x_1 x_2 \ldots x_{n-1} x_n})^2 = \\
&= 10^{2n-2} x_1^2 + 10^{2n-4} x_2 \cdot \underline{\overline{(2x_1)\, x_2}} + 10^{2n-6} x_3 \cdot \underline{\overline{(2 \cdot (\overline{x_1 x_2}))\, x_3}} + \cdots + x_n \cdot \underline{\overline{(2 \cdot (\overline{x_1 x_2 \ldots x_{n-1}}))\, x_n}}
\end{aligned}
$$

Dokaz: Po izreku o kvadratu n-členika lahko poskusimo z dokazom izreka.

Poskusimo preoblikovati levo stran v desno.

$$
\begin{aligned}
&(\overline{x_1 x_2 \ldots x_{n-1} x_n})^2 = \\
&= \left(10^{n-1} \cdot x_1 + 10^{n-2} \cdot x_2 + \cdots + 10^1 \cdot x_{n-1} + x_n\right)^2 = \\
&= 10^{2n-2} \cdot x_1^2 + 10^{2n-4} \cdot x_2^2 + \cdots + 10^2 \cdot x_{n-1}^2 + x_n^2 + \\
&\quad + 2 \cdot 10^{2n-3} x_1 x_2 + 2 \cdot 10^{2n-4} x_1 x_3 + \cdots + 2 \cdot 10^{n-1} x_1 x_n + \\
&\quad + 2 \cdot 10^{2n-5} x_2 x_3 + 2 \cdot 10^{2n-6} x_2 x_4 + \cdots + 2 \cdot 10^{n-2} x_2 x_n + \\
&\quad + \cdots + 2 \cdot 10^1 x_{n-1} x_n
\end{aligned}
$$

Sedaj moramo spretno družiti in izpostavljati. Člene družimo tako, da izpostavimo enako desetiško potenco, izpostavimo vse, kar se da. Poleg tega v i-tem koraku družimo člene, ki vsebujejo $x_1, x_2 \ldots x_i$. Pri tem nas vodi prvi del izraza vsote kvadratov števk.

$$
\begin{aligned}
&= 10^{2n-2} x_1^2 + 10^{2n-4} x_2 \cdot (20x_1 + x_2) + \\
&\quad + 10^{2n-6} x_3 \cdot (200x_1 + 20x_2 + x_3) + \cdots + \\
&\quad + x_n \cdot \left(2 \cdot 10^{n-1} x_1 + 2 \cdot 10^{n-2} x_2 + \cdots + 2 \cdot 10 \cdot x_{n-1} + x_n\right) = \\
&= 10^{2n-2} x_1^2 + 10^{2n-4} x_2 \cdot \underline{\overline{(2x_1)\, x_2}} + 10^{2n-6} x_3 \cdot \underline{\overline{(2 \cdot (\overline{x_1 x_2}))\, x_3}} + \\
&\quad + \cdots + x_n \cdot \underline{\overline{(2 \cdot (\overline{x_1 x_2 \ldots x_{n-1}}))\, x_n}}
\end{aligned}
$$

□

Komentirajmo še nekaj stvari, ki smo jih na prejšnjih straneh že utemeljili, vendar takrat pri konkretnih n, sedaj pa v splošnem.

- **Razdelitev na bloka po dva**

  Korenjenec razdelimo na bloke po dve števki. Vemo že, da brez škode za splošnost to lahko storimo, ne glede na to če je korenjenec sodo- ali lihomesten $\overline{A_1 d_1 \dots d_{2n-3} d_{2n-2}}$.

  Blok A_1 pravzaprav snuje vrednost $10^{2n-2} \cdot \left\lfloor \frac{A_1}{10^{2n-2}} \right\rfloor$. To vrednost pa v največji možni meri zavzamemo z izrazom $10^{2n-2} x_1^2$. Razlika do prave vrednosti korenjenca $10^{2n-2} \cdot O_1$, ki je ne zavzamemo, in pridatek $10^{2n-4} \cdot \overline{d_1 d_2}$ pa zavzamemo z vrednostjo $10^{2n-4} \cdot \overline{(2x_1)x_2} \cdot x_2$. Izraz $10^{2n-2} x_1^2$ je neodvisen od $10^{2n-4} \cdot \overline{d_1 d_2}$.

  Enako sklepanje lahko naredimo za vsak preostali blok A_i, kjer ta snuje vrednost $10^{2n-2} \left\lfloor \frac{A_1}{10^{2(n-i)}} \right\rfloor$, zato je tudi tu smiselna razdelitev na bloke po dve števki.

- **Enolična določenost prve števke**

  Algoritem veleva, da število v prvem bloku z leve korenimo in v rezultat pišemo spodnji celi del tega korena. Vprašanje je, zakaj nobeno manjše oziroma večje celo število ne pride v poštev.

  Imejmo število $\overline{A_1 d_1 \dots d_{2n-3} d_{2n-2}}$, kjer so d_i števke. Oznaka A_1 pa je v splošnem števka posplošenega desetiškega sestava, ki zastopa prvi blok, to pa je bodisi enomestno bodisi dvomestno naravno število.

  Algoritem pravi, da velja

  $$x_1 = \left\lfloor \sqrt{A_1} \right\rfloor,$$

  kjer z oglatima uklepajem in zaklepajem označimo spodnji celi del korena. Po točki b) izreka 4.2 to pomeni

  $$\sqrt{A_1} - 1 < x_1 \leq \sqrt{A_1}$$

  oziroma

  $$\left(\sqrt{A_1} - 1 \right)^2 < x_1^2 \leq A_1.$$

  S protislovjem dokažimo, da je tak x_1 edini možen. Ločimo dve možnosti:

  1. Namesto x_1, za katerega velja zgornji pogoj, vzemimo $x_1 - 1$.

     Če dokažemo, da $x_1 - 1$ ni ustrezen, potem tudi nobena manjša vrednost za x_1 ne bo. Zdaj preverimo, ali lahko s kvadriranjem tudi zdaj nekako dosežemo prvotni kvadrat. V mejnem primeru za vse ostale števke števila, ki ga kvadriramo, določimo, da so največje možne, torej 9.

     $$\overline{(x_1 - 1)\underbrace{9 \dots 9}_{(n-1)\text{-krat}}} = \overline{x_1 \underbrace{0 \dots 0}_{(n-1)\text{-krat}}} - 1$$

$$\left(\overline{(x_1-1)\underbrace{9\dots 9}_{(n-1)\text{-krat}}}\right)^2 = \left(\overline{x_1\underbrace{0\dots 0}_{(n-1)\text{-krat}}}-1\right)^2 < \overline{x_1^2\underbrace{0\dots 0}_{(2n-2)\text{-krat}}} \le \overline{A_1\underbrace{0\ \dots 0}_{(2n-2)\text{-krat}}}$$
$$\le \overline{A_1 d_1 \dots d_{2n-3} d_{2n-2}}$$

Tak izbor za prvo števko rezultata očitno ne ustreza.

2. Namesto x_1, za katerega velja zgornji pogoj, vzemimo x_1+1.
Če dokažemo, da x_1+1 ni ustrezen, potem tudi nobena večja vrednost za x_1 ne bo. Zdaj preverimo, ali lahko s kvadriranjem tudi zdaj nekako še ujamemo prvotni kvadrat. V mejnem primeru za ostale števke števila, ki ga kvadriramo, določimo, da so najmanjše možne, torej 0.

$$\left(\overline{(x_1+1)\underbrace{0\dots 0}_{(n-1)\text{-krat}}}\right)^2 = 10^{2n-2}\left(x_1+1\right)^2 >$$
$$> 10^{2n-2}(A_1+1)-1 = 10^{2n-2}A_1 + \underbrace{9\dots 9}_{(2n-2)\text{-krat}} \ge$$
$$\ge \overline{A_1\,\underbrace{9\ \dots 9}_{(2n-2)\text{-krat}}} \quad \ge$$
$$\ge \overline{A_1 d_1 \dots d_{2n-3} d_{2n-2}}$$

Prvo neenakost v zgornji vrstici utemeljimo na podlagi sledečega dejstva: Ker je A_1 celo število, velja očitna zveza $\lfloor\sqrt{A_1}\rfloor = \left\lfloor\sqrt{A_1+99/100}\right\rfloor$. Vemo že, da je

$$\sqrt{A_1}-1 < x_1 \le \sqrt{A_1},\text{ torej } \sqrt{A_1}-1 < x_1,$$

od koder sledi

$$\sqrt{A_1} < x_1+1$$

in od tod

$$A_1 < \left(x_1+1\right)^2.$$

Na podlagi zadnjih ugotovitev lahko zapišemo:

$$\begin{aligned}
& x_1 = \left\lfloor\sqrt{A_1+99/100}\right\rfloor \\
\Longrightarrow\quad & \sqrt{A_1+99/100}-1 < x_1 \le \sqrt{A_1+99/100} \\
\Longrightarrow\quad & A_1+99/100 < \left(x_1+1\right)^2 \\
\Longrightarrow\quad & 100\left(x_1+1\right)^2 > 100A_1+99 \\
\Longrightarrow\quad & 100\left(x_1+1\right)^2 > 100\left(A_1+1\right)-1.
\end{aligned}$$

Tudi tak izbor za prvo števko rezultata očitno ne ustreza.

- **Prvi ostanek**

Ko določimo x_1, v algoritmu x_1 kvadriramo in ga podpišemo pod prvi blok ter odštejemo pa dobimo ostanek O_1. V resnici pa, kar razjasni razpisan kvadrat dvomestnega števila, od prvotnega števila odštejemo $10^{2n-2}x_1^2$. Nadaljnje števke v rezultatu morajo zajeti vrednost tega ostanka, saj je od njih odvisen.

- **Množenje z 2 in dopisane neprve števke x_i**

  V algoritmu se vsakič, ko določamo i-to števko, vprašamo: *"Katera je največja števka, ki je lahko dopisana dvakratniku dosedanjega rezultata, da dobljeno število, pomnoženo s to števko, da rezultat, ki je manjši ali enak od zahtevka?"*

  Slednje prav tako utemelji izrek 5.2.

  Ostanku O_{i-1} pripišemo blok A_i, tako dobljeni zahtevek $B_i = \overline{O_{i-1}0A_i}$ izrazimo z izrazom

$$10^{2(n-i-1)} \cdot x_{i+1} \cdot \overline{(2 \cdot \overline{x_1 x_2 \dots x_i})\, x_{i+1}},$$

  ki pa je pravzaprav identičen navodilu oziroma vprašanju: *"Katera je največja števka x_{i+1}, ki je lahko dopisana številu $(2 \cdot \overline{x_1 x_2 \dots x_i})$, da bo dobljeno število, pomnoženo s to števko, manjše ali enako številu zahtevka?"* Faktor $10^{2(n-i-1)}$ je smiselen, ker blok A_{i+1} v resnici predstavlja $10^{2(n-i-1)} A_{i+1}$.

  Ker obravnavamo koren kvadrata n-mestnega števila, vemo, da se bo korenjenje izšlo. Tako lahko rečemo, da iščemo tako števko x_n, da je vrednost C_n enaka zahtevku B_n.

  Razmislimo še, zakaj mora biti ne prva x_i največja taka, da je $B_i \geq C_i$. Denimo, da je x_i največja ustrezna števka. Poglejmo, kaj se zgodi, če namesto nje napišemo števko $x_i - 1$, vse nadaljnje števke pa zato naredimo kar največje možne, torej 9. V dokaz, da mora biti števka x_3 največja možna, se moramo prepričati, da je

$$(\overline{x_1 x_2 \dots x_n})^2 > \Big(\overline{x_1 \dots x_{i-1}(x_i - 1)\underbrace{9 \dots 9}_{(n-i)\text{-krat}}}\Big)^2$$

  ali ekvivalentno

$$\overline{x_1 x_2 \dots x_n} > \overline{x_1 \dots x_{i-1}(x_i - 1)\underbrace{9 \dots 9}_{(n-i)\text{-krat}}}.$$

  Pišemo

$$\overline{x_1 x_2 \dots x_n} - \overline{x_1 \dots x_{i-1}(x_i - 1)\underbrace{9 \dots 9}_{(n-i)\text{-krat}}} = 1 + \overline{x_{i+1} x_{i+2} \dots x_n} > 0$$

  saj je $\overline{x_{i+1} x_{i+2} \dots x_n} \geq 0$. S tem smo hoteno utemeljili.

  Podobno razmišljamo tudi pri utemeljitvi, zakaj števka x_i ne more biti večja od največje take, za katero velja $B_i \geq C_i$. Tokrat namesto x_i računamo z $x_i + 1$, vse števke za pa naj bodo kar najmanjše možne, torej 0. Dokazati moramo, da je

$$\overline{x_1 x_2 \dots x_n} < \overline{x_1 \dots x_{i-1}(x_i + 1)\underbrace{0 \dots 0}_{(n-i)\text{-krat}}}.$$

  Pišemo

$$\overline{x_1 \dots x_{i-1}(x_i + 1)\underbrace{0 \dots 0}_{(n-i)\text{-krat}}} - \overline{x_1 x_2 \dots x_n} = 10^{n-i} - \overline{x_{i+1} x_{i+2} \dots x_n} > 0,$$

  saj je $10^{n-i} > \overline{x_{i+1} x_{i+2} \dots x_n}$. S tem smo hoteno utemeljili.

- **Neprvi ostanki**

  Ko določimo x_i, i-ti ostanek določimo kot $O_i = B_i - C_i$. V skladu s prejšnjo točko nato določimo števko x_{i+1}. Ker korenimo kvadrat n-mestnega naravnega števila, jasno velja $O_n = B_n - C_n = 0$.

6.3.4 Decimalna števila

Na matematičnem forumu [**1**] je bilo korenjeno decimalno število. Utemeljiti gre, kako lahko korenimo decimalna števila.

Decimalno število D korenimo tako, da ga pomnožimo z desetiško potenco 10^{2k} za nek $k \in \mathbb{N}$, tako da se znebimo decimalne vejice in dobimo naravno število N.

Dobljeno naravno število N znamo koreniti. To izgleda tako:

$$\sqrt{N} = \sqrt{D \cdot 10^{2k}} = \sqrt{D} \cdot \sqrt{10^{2k}} = \sqrt{D} \cdot 10^k.$$

Da dobimo prvotno decimalno število D, moramo pri korenjenju dobljeno le še deliti z 10^k.

$$\sqrt{D} = \frac{\sqrt{N}}{10^k}$$

Na podlagi izpeljave lahko sklepamo, da če se po vseh porabljenih blokih postopek ne konča z 0, postopek nadaljujemo, tako da k številu dodamo blok 00, ki nam bo zagotovil novo decimalno mesto rezultata. Postopek ponavljamo, dokler kot ostanek ne dobimo ničle oziroma do željene natančnosti.

Da se izognemo dodatnemu preračunavanju, korenjenje decimalnih števil lahko zvežemo v en algoritem, tako da v algoritem za ročno korenjenje vključimo decimalno vejico. Na začetku algoritma število od decimalne vejice razdelimo v bloke po dve mesti (števki) levo in desno. Če ima celi del števila liho mest − nič za to, tako kot prio korenjenju naravnih števil bo imel prvi blok z leve le en. mesto Če pa je decimalnih mest korenjenca liho mnogo, moramo v skrajni desni blok dodati števko 0. Temu je tako, saj smo v zgornji izpeljavi korenjenec množili z 10^{2k}, da smo po korenjenju lahko rezultat korenjenja pomnožili z desetiško potenco 10^k, $k \in \mathbb{N}$. Z drugimi besedami, število decimalk korenjenca mora biti sodo, za kar lahko brez škode poskrbimo tako, da v primeru lihega števila decimalnih mest na konec števila dodamo števko 0.

***

Do izbranega decimalnega mesta lahko računamo tudi korene nepopolnih kvadratov − iracionalna števila.

Primer: Ugotavljamo lahko decimalke števila $\sqrt{2}$. Ugotovimo na primer $\sqrt{2}$ na 5 decimalk natančno. Preglednosti v čast bomo v tem primeru korenili z daljšo različico postopka.

Če nas zanima 5 decimalnih mest, bomo vsled tega številu 2 v algoritmu dopisali 5 blokov po 00. En dodatni blok korenjenca namreč snuje eno mesto rezultata. Korenimo torej število 20000000000.

1.

$$\begin{array}{l} \sqrt{2|00|00|00|00|00} = 1 \\ \underline{1} \\ 1\ 00 \end{array}$$

2.

$$\begin{array}{l}
\sqrt{2|00|00|00|00|00} = 14 \\
\underline{1} \\
1\ 00 \models 2\underline{4} \cdot \underline{4} \\
\quad\underline{96} \\
\qquad 4\ 00
\end{array}$$

3.

$$\begin{array}{l}
\sqrt{2|00|00|00|00|00} = 141 \\
\underline{1} \\
1\ 00 \models 2\underline{4} \cdot \underline{4} \\
\quad\underline{96} \\
\qquad 4\ \ 00 \models 28\underline{1} \cdot \underline{1} \\
\qquad \underline{2\ \ 81} \\
\qquad 1\ \ 19\ \ 00
\end{array}$$

4.

$$\begin{array}{l}
\sqrt{2|00|00|00|00|00} = 1414 \\
\underline{1} \\
1\ 00 \models 2\underline{4} \cdot \underline{4} \\
\quad\underline{96} \\
\qquad 4\ \ 00 \models 28\underline{1} \cdot \underline{1} \\
\qquad \underline{2\ \ 81} \\
\qquad 1\ \ 19\ \ 00 \models 282\underline{4} \cdot \underline{4} \\
\qquad \underline{1\ \ 12\ \ 96} \\
\qquad\qquad 6\ \ 04\ \ 00
\end{array}$$

5.

$$\begin{array}{l}
\sqrt{2|00|00|00|00|00} = 14142 \\
\underline{1} \\
1\ 00 \models 2\underline{4} \cdot \underline{4} \\
\quad\underline{96} \\
\qquad 4\ \ 00 \models 28\underline{1} \cdot \underline{1} \\
\qquad \underline{2\ \ 81} \\
\qquad 1\ \ 19\ \ 00 \models 282\underline{4} \cdot \underline{4} \\
\qquad \underline{1\ \ 12\ \ 96} \\
\qquad\qquad 6\ \ 04\ \ 00 \models 2828\underline{2} \cdot \underline{2} \\
\qquad\qquad \underline{5\ \ 65\ \ 64} \\
\qquad\qquad\quad 38\ \ 36\ \ 00
\end{array}$$

6.

$$
\begin{array}{l}
\sqrt{2|00|00|00|00|00} = 141421 \\
\underline{1} \\
1\ 00 \models 2\underline{4} \cdot \underline{4} \\
\quad \underline{96} \\
\qquad 4\ 00 \models 28\underline{1} \cdot \underline{1} \\
\qquad \underline{2\ 81} \\
\qquad 1\ 19\ \ 00 \models 282\underline{4} \cdot \underline{4} \\
\qquad \underline{1\ 12\ 96} \\
\qquad\qquad 6\ 04\ \ 00 \models 2828\underline{2} \cdot \underline{2} \\
\qquad\qquad \underline{5\ 65\ 64} \\
\qquad\qquad\quad 38\ 36\ \ 00 \models 28284\underline{1} \cdot \underline{1} \\
\qquad\qquad\quad \underline{28\ 28\ 41} \\
\qquad\qquad\quad 10\ 07\ 59
\end{array}
$$

Dobili smo 141421. Ker smo število 2 množili z $10^{2 \cdot 5} = 10^{10}$, je potem:

$$\sqrt{2} \doteq \frac{\sqrt{2 \cdot 10^{10}}}{10^5} = \frac{141421}{10^5} = 1,41421.$$

In res, v tabelah lahko poiščemo tudi natančnejši približek $\sqrt{2} \doteq 1,41421356237$, pri čemer pa vidimo, da se prvih pet decimalnih mest zares ujema s tistimi, ki jih daje algoritem za ročno korenjenje.

V komentar pa je treba dodati, da zaradi vedno večjih števil, ki nastopajo v računih algoritma, koraki postajajo vedno daljši in miselno napornejši. Tako da je bi bilo do 11. decimalke z računanjem na roke že kar naporno priti. Kljub temu pa je v teoretičnem smislu decimalke moč računati v nedogled.

7 Kubični koren

Utemeljili smo postopek za kvadratno korenjenje. Kar hitro pa se nadalje pojavi vprašanje, ali je obstaja tudi algoritem za kubično korenjenje.

Glede na že poznanega za kvadratno korenjenje ga lahko ugibamo; takšna so naša ugibanja. Algoritem je enak kot tisti za kvadratno korenjenje, razlike so le te:

1. Namesto v bloke po dve števki bomo števke družili v bloke po tri.
2. Nov izraz za **vrednost** pri tretjem korenjenju C_i^3, kjer 3 na mestu eksponenta ne pomeni potenciranja, temveč je zaradi preglednosti indeks, ki označuje, da gre za vrednost pri kubičnem korenjenju.

Vsi ostali deli algoritma, ki smo jih definirali na strani 15, zato uvedba novih oznak zanje ni potrebna.

Algoritem smo pri kvadratnem korenu utemeljili s spretnim razstavljanem kvadrata števila, ki ga s korenjenjem dobimo. Postopajmo podobno.

7.1 Kub dvomestnega naravnega števila

Predstavljajmo si kub dvomestnega naravnega števila. Dajmo kar splošno.

$$(\overline{x_1x_2})^3 = (10x_1 + x_2)^3 = 1000x_1^3 + 300x_1^2x_2 + 30x_1x_2^2 + x_1^3$$

Prvi člen na levi opisuje prvi korak algoritma. Ko določimo x_1, potem moramo v drugem koraku določiti x_2.

$$\begin{aligned}(\overline{x_1x_2})^3 = (10x_1 + x_1)^3 = 1000x_1^3 + 300x_1^2x_2 + 30x_1x_2 + x_2^3\\ 1000x_1^3 + x_2 \cdot \left(300x_1^2 + 30x_1x_2 + x_2^2\right)\end{aligned}$$

Izraz $300x_1^2 + 30x_1x_2 + x_2^2$ je izraz za drugo **vrednost** C_2^3 pri algoritmu za kubično korenjenje. Poglejmo, ali ga moremo, podobno kot tistega pri kvadratnemu korenu, spretno in uporabno razstaviti. Temu je tako takrat, ko je diskriminanta večja od nič, $D > 0$. V izrazu zgoraj označimo $a = 300, b = 30x_2, c = x_2^2$. Računamo diskriminanto …

$$D = b^2 - 4ac = (30x_2)^2 - 4 \cdot 300 \cdot x_2^2 = 900x_2^2 - 1200x_2^2 = -300x_2^2 < 0.$$

Torej se vrednosti za kubično korenjenje $C_2^3 = 300x_1^2 + 30x_1x_2 + x_2^2$ ne da razstaviti.

Vseeno poskusimo izraz zapisati v čim bolj prikladni obliki. Z nekaj truda ugotovimo sledeče:

$$\begin{aligned}(\overline{x_1x_2})^3 &= 1000x_1^3 + x_2 \cdot \left(300x_1^2 + 30x_1x_2 + x_2^2\right)\\ &= 1000x_1^3 + 30x_1x_2\,(10x_1 + x_2) + x_2^3\\ &= 1000x_1^3 + 30x_1x_2 \cdot \overline{x_1x_2} + x_2^3.\end{aligned}$$

***

Poskusimo s to obliko prvotnega izraza koreniti kub dvomestnega naravnega števila. Tako je na primer število 175616. Zaradi preglednosti se bomo v simbolih za določene vrednosti algoritma odrekli indeksom 3, ki smo jih pisali nad simboli.

1. Število razdelimo na dva bloke po tri števke. Vprašamo se: *“Katero je največje naravno število x_1, za katerega velja, da je njegov kub manjši od 175?”* To je jasno število $x_1 = 5$, saj je $5^3 < 175$. Desno od enačaja pišemo prvi rezultat $R_1 = 5$, pod prvi blok $A_1 = 175$ pa $5^3 = 125$. Odštejemo, pod črto zapišemo prvi ostanek, se pravi $O_1 = 175 - 125 = 50$. Prepišemo naslednji blok $A_2 = 616$, da dobimo zahtevek B_2.

$$\begin{array}{l} \sqrt[3]{175|616} = 5 \\ \underline{125} \\ \quad 50\,616 \models \end{array}$$

2. Zahtevek B_2 se mora kar najbolj prileči izrazu za vrednost C_2^3 pri kubičnem korenjenju. Na polja s črticami je treba vpisati največje tako števko, da je $B_2 \geq C_2^3$.

$$\begin{array}{l} \sqrt[3]{175|616} = 5 \\ \underline{125} \\ \quad 50\,616 \models 30 \cdot 5 \cdot \_ \cdot \overline{5\_} + \_^3 \end{array}$$

3. Izraz za vrednost C_i^3 je slabo razstavljen, saj se ga zares razstaviti sploh ne da. Le s površnim ugibanjem lahko najdemo števko, ki jo gre vpisati na črte. Ta števka je 6, veliko lažje kot iz vrednosti jo sklepamo iz zadnje števke korenjenca ali zahtevka, ki je 6. Torej je $x_2 = 6$, rezultat pa $R_2 = 56$.

$$\begin{array}{l} \sqrt[3]{175|616} = 56 \\ \underline{125} \\ \quad 50\,616 \models 30 \cdot 5 \cdot \underline{6} \cdot \overline{5\underline{6}} + \underline{6}^3 \\ \quad \underline{50\,616} \\ \qquad\quad 0 \end{array}$$

Odštejemo, dobimo ostanek $O_2 = 0$. Torej smo zaključili.

***

Algoritem očitno deluje. Opazili smo, da je vrednost težko oceniti že v drugem koraku, saj števka x_2, ki smo jo morali poiskati, nastopa prevečkrat in v preveč različnih oblikah. Stvar se še dodatno zaplete v nadaljnih korakih.

Vsled vsega tega algoritem za kubično korenjenje nima za človeka praktične vrednosti, toda ob *matematični* predpostavki, da smo zmožni hitrega računanja osnovnih operacij oziroma imamo na voljo tako žepno računalo, s tem pa zmožni iz vrednosti C_i^3 ocenjevati števke x_i, je algoritem povsem korekten in uporaben, saj deluje.

Ob upoštevanju take predpostavke je spretno razstavljanje izraza za vrednost pravzaprav nesmiselno, tako da lahko z nekaj poguma postavimo predvidevanje

$$C_i^3 = x_i \cdot \left(300 \cdot (\overline{x_1 x_2 \dots x_{i-1}})^2 + 30 \cdot (\overline{x_1 x_2 \dots x_{i-1}})\, x_i + x_i^2\right).$$

Algoritem bi morali utemeljiti še v vsej splošnosti, saj smo zdaj pokazali le za primer tretjega korenjenja kubov dvomestnih števil. Vendar je dokaz, pri katerem bi morali kubirati n-členik, preprosto preveč obsežen, natančno utemeljevanje algoritma za kubično korenenje pa ni v domeni te raziskovalne naloge. Zaenkrat se zadovoljimo z algoritmom, ki smo ga na podlagi algoritma za ročno kvadratno korenjenje in algoritma za kubično korenjenje kubov dvomestnih števil predvideli in delno utemeljili.

8 Splošni koren, koren k-te stopnje

Zamika nas, da bi pogledali oziroma ugotovili, kakšen je algoritem za splošno korenjenje s korenskim eksponentom k, ali lepše: k-to korenjenje .

Glede na to da že poznamo algoritma za kvadratno in (vsaj do neke mere) kubično korenjenje, ga lahko ugibamo. Algoritem je enak kot tisti za kvadratno korenjenje, razlike so le te:

1. Namesto v bloke po dve števki bomo števke družili v bloke po k.
2. Nov izraz za **vrednost** pri k-tem korenjenju

$$C_i^k = \left(\overline{(\overline{x_1 x_2 \dots x_{i-1}})\, x_i}\right)^n - 10^n \left(\overline{x_1 x_2 \dots x_{i-1}}\right)^n$$

ki je zagotovo veliko težje razstavljiv kot izraz C_i^3, ki nam ga ni uspelo razstaviti. Vsled tega težko govorimo o uporabni vrednosti takega algoritma za k-to korenjenje, vseeno pa ostajamo zadovoljni z matematičnega stališča.

***

Vendar lahko nekatere konstrukcije algoritmov za k-to korenjenje poiščemo tudi na nekoliko bolj zvit način. Ključna lastnost, ki nam to omogoča, je

$$\sqrt[m \cdot n]{a} = \sqrt[m]{\sqrt[n]{a}},$$

ki smo jo že pokazali v teoretičnem uvodu. Zaradi nje lahko trdimo, da bi morali za vso splošnost zgolj najti algoritem za ročno korenjenje le za praštevilske korenske eksponente in se s tem nekaj ostalim algoritmom lahko izognemo.

Ker do zdaj poznamo le algoritma za ročno računanje 2. in 3. korena, lahko v sled tega enostavno korenimo tudi s na primer 6. korenom, saj je

$$\sqrt[6]{a} = \sqrt[2 \cdot 3]{a} = \sqrt[3]{\sqrt[2]{a}}.$$

Na podoben način lahko ročno računamo vse korene, katerih korensko stopnjo lahko zapišemo v obliki

$$2^{l_1} \cdot 3^{l_2} \text{ za vse } l_1, l_2 \in \mathbb{N} \cup \{0\}.$$

Tako lahko z našima algoritmoma računamo četrte, osme, devete korene ... in tudi na primer koren stopnje 5184.

Poskusimo ugotovljeno uporabiti in 6-to koreniti število 5489031744. Število bomo najprej kvadratno korenili, nato pa še kubično.

1.

$$\begin{array}{l}
\sqrt{54|89|03|17|44} = 7 \\
\ \ \underline{49} \\
\ \ \ \ 5\ 89
\end{array}$$

2.

$$\begin{array}{l}
\sqrt{54|89|03|17|44} = 74 \\
\ \ \underline{49} \\
\ \ \ \ 5\ 89 \models 14\underline{4} \cdot \underline{4} \\
\ \ \ \ \underline{5\ 76} \\
\ \ \ \ \ \ 13\ 03
\end{array}$$

3.

$$\begin{array}{l}
\sqrt{54|89|03|17|44} = 740 \\
\ \ \underline{49} \\
\ \ \ \ 5\ 89 \models 14\underline{4} \cdot \underline{4} \\
\ \ \ \ \underline{5\ 76} \\
\ \ \ \ \ \ 13\ 03 \models 148\underline{0} \cdot \underline{0} \\
\ \ \ \ \ \ \ \ \ \ \ \ \underline{0} \\
\ \ \ \ \ \ 13\ 03\ 17
\end{array}$$

4.

$$\begin{array}{l}
\sqrt{54|89|03|17|44} = 7408 \\
\ \ \underline{49} \\
\ \ \ \ 5\ 89 \models 14\underline{4} \cdot \underline{4} \\
\ \ \ \ \underline{5\ 76} \\
\ \ \ \ \ \ 13\ 03 \models 148\underline{0} \cdot \underline{0} \\
\ \ \ \ \ \ \ \ \ \ \ \ \underline{0} \\
\ \ \ \ \ \ 13\ 03\ 17 \models 1480\underline{8} \cdot \underline{8} \\
\ \ \ \ \ \ \underline{11\ 84\ 64} \\
\ \ \ \ \ \ \ \ 1\ 18\ 53\ 44
\end{array}$$

5.

$$
\begin{array}{l}
\sqrt{54|89|03|17|44} = 74088 \\
\underline{49} \\
\quad 5\ 89 \models 14\underline{4} \cdot \underline{4} \\
\quad \underline{5\ 76} \\
\qquad 13\ 03 \models 148\underline{0} \cdot \underline{0} \\
\qquad\qquad \underline{0} \\
\qquad 13\ 03\ 17 \models 1480\underline{8} \cdot \underline{8} \\
\qquad \underline{11\ 84\ 64} \\
\qquad\ \ 1\ 18\ 53\ 44 \models 14808 \cdot \underline{8} \\
\qquad\ \ \underline{1\ 18\ 53\ 44} \\
\qquad\qquad\qquad\quad 0
\end{array}
$$

Zdaj ko imamo kvadratni koren, moramo najti še kubični koren le-tega.

1.

$$
\begin{array}{l}
\sqrt[3]{74|088} = 4 \\
\quad \underline{64} \\
\quad 10\ 088 \models
\end{array}
$$

2.

$$
\begin{array}{l}
\sqrt[3]{74|088} = 42 \\
\quad \underline{64} \\
\quad 10\ 088 \models 30 \cdot 4 \cdot \underline{2} \cdot \overline{4\underline{2}} + \underline{2}^3 \\
\quad \underline{10\ 088} \\
\qquad\quad 0
\end{array}
$$

Naš odgovor je torej 42 − torej odgovor na ultimativno vprašanje o Življenju, Vesolju in sploh vsem! Poleg tega pa zares znamo še 6-to koreniti!

9 Zaključek

V vsej matematični natančnosti smo dokazali algoritem za kvadratno, delno tudi za kubično korenjenje. Podali smo tudi nekaj ugotovitev glede splošnega algoritma za k-to korenjenje, ki pa ga nismo niti skušali utemeljiti matematično korektno. Takšen izziv je morda lahko ideja za novo raziskovalno nalogo. Vtis je, da bi bila ta precej kompleksnejša z morda manj zanimivimi in v praksi uporabnimi izsledki.

Zavedati se moramo, da posebej pri večjih, to je mnogo mestnih številih, tudi pri ročnem korenjenju hitro pridemo v situacijo, ko si račune želimo izvajati s pomočjo računala. S tega vidika se na prvi pogled morda zdi, da naš algoritem nima več neke posebne vrednosti, saj ročno korenjenje sploh ni več ročno, temveč tudi pri njem uporabljamo tehnologijo.

Vendar pa opazimo, da smo, čeprav morda uporabimo računalo, operacijo korenjenja reducirali na osnovnejši operaciji množenja in odštevanja. S tega vidika je postopek zanimiv povsem teoretično.

Literatura

[1] Matemanija. (21. 5. 2013). *„Ručno" izračunavanje kvadratnog korena.* Pridobljeno 30. 6. 2018 s http://forum.matemanija.com/viewtopic.php?f=46&t=317.

[2] Cedilnik, A. (2006) *Matematični priročnik.* Radovljica: Didakta.