Računalniška uporaba algoritma za mehanske razcepe števil

Eva ZUPAN

13. marec 2002

1 Uvod

S preprostim računom lahko izračunamo produkt poljubno velikih naravnih števil, če pa imamo dano veliko število ter iščemo njegove prafaktorje, je naloga precej težja. Zahtevnost se še posebej poveča, če je število produkt dveh velikih praštevil. Razcep velikih števil je bil že v preteklosti izziv za mnoge matematike.

Pred 200 leti je deset-mestno število veljalo za veliko, težko razcepno število. Deset in več-mestna števila, ki so produkt dveh velikih praštevil, pa skoraj ni bilo mogoče razcepiti. Ko so se naloge lotili z računalniško podporo je izraz veliko število dobil popolnoma nove razsežnosti. Ker so tudi zmogljivosti računalnikov omejene, se je napredek ustavil, zato so začeli iskati nove algoritme, ki bi skrajšali časovno zahtevnost naloge. Eden uspešnejših algoritmov za razcep števil sloni na uporabi eliptičnih krivulj. Prvi je to metodo opisal H. W. Lenstra, Jr. v publikaciji '*Factoring integers with elliptic curves*' (glej [5], str. 209 in 213). H. Dubner je leta 1986 s pomočjo te metode izračunal 18-mesten faktor 116-mestnega števila $10^{142} + 1$. Zaradi sodobnih kriptografskih metod, ki slonijo na produktih velikih praštevil (npr. RSA, glej [3], [5]), je razbijanje na prafaktorje zelo aktualna naloga, s katero se ukvarja mnogo ljudi. Nekaj aktualnih 'rekordov' v iskanju velikih praštevil in razcepov povzemamo po spletnih straneh [8] in [9]. Največje do sedaj znano praštevilo je $2^{6\,972\,593} - 1$, ki ima 2 098 960 decimalnih mest. Največje razcepljeno sestavljeno število posebne oblike je $2^{773} + 1$ in ima 233 decimalnih mest, največje običajno razcepljeno število pa ima 'samo' 155 decimalnih mest.

Področje teorije števil je bilo v preteklosti precej raziskano, vendar je brez računalnikov preverjanje potekalo zgolj na majhnih številih. Zato je bila pomembna ugotovitev, da lahko algoritem razcepa zapišemo v obliki primerni za avtomatizacijo. Tako lahko iskanje razcepa števil prevedemo v obliko, ki jo rešujemo z mehanskim strojem. Prvi, ki je odkril možnost mehanskega iskanja razcepa naravnih števil, je bil britanski matematik Frederick William Lawrence.

Svoja dognanja je objavil leta 1896. Prevod članka v francoščino, ki ga je leta 1910 André Gérardin izdal v reviji *Sphinx-Oedipe* je spodbudil več matematikov, da so poskusili sestaviti mehanski stroj. Leta 1912 sta se s konstrukcijo stroja začneta ukvarjati tudi brata Pierre in Eugène Oliver Carissan. Idejo stroja je Pierre predstavil v reviji *Sphinx-Oedipe*, medtem ko ga je brat Eugène Oliver sestavil. Podrobnih podatkov o stroju ni, verjetno pa ni bil preveč učinkovit. Prvi znani računski stroj, ki je uspešno razbijal velika cela števila na prafaktorje je bil zgrajen med leti 1913-1919 po načrtih E. O. Carissana in se imenuje '*machine a congruence*', oziroma '*kongruenčni stroj*'. Nato je bil popolnoma pozabljen 75 let. Šele leta 1992 so J. Shallit, H. C. Williams in F. Morain stroj spet našli in ga predstavili v članku [1] in na spletu [7].

V članku podrobneje predstavimo matematično ozadje, na osnovi katerega je deloval kongruenčni stroj. Opis stroja povzemamo po [1], dodajamo pa podrobnejšo razlago sestave in delovanja stroja. Ker stroja seveda nismo mogli testirati, od leta 1994 ga hrani *Conservatoire Nationale des Arts et Métiers* v Parizu, smo delovanje stroja simulirali z računalniškim programom.

2 Matematično ozadje

Vsako sodo število znamo razcepiti, zato se omejimo na razcep lihih števil. Nakatere algebrske lastnosti nam zagotavljajo, da so števila, ki jih lahko zapišemo v specifični obliki, razcepna. Naštejmo nekaj idej za razcep števil:

- Če lahko število n zapišemo kot vsoto ali razliko kubov dveh števil, ga lahko zapišemo tudi kot produkt:

$$n = x^3 \pm y^3 = (x \pm y)\left(x^2 \mp xy + y^2\right).$$

- Število n je razcepno, če ga lahko zapišemo na dva različna načina kot vsoto kvadratov dveh števil (glej [2]).

$$n = x^2 + y^2 = x_1^2 + y_1^2$$

  Če dva taka različna zapisa ne obstajata, je število nerazcepno oz. praštevilo.

- Število n je razcepno, če ga lahko zapišemo kot razliko kvadratov dveh števil.

$$n = x^2 - y^2 = (x - y)(x + y).$$

  Podrobneje si oglejmo razcep števil, ki temelji na razliki kvadratov. V [5] pa si lahko ogledamo kako je isto idejo porabil Fermat za ročno faktorizacijo.

Trditev 1 *Naj bo n naravno liho sestavljeno število. Potem obstajata taki celi števili x in y, $x \geq 0$ in $y \geq 0$, da velja*

$$n = x^2 - y^2 = (x - y)(x + y)$$

in $(x - y)(x + y)$ ni trivialen razcep.

Dokaz: Recimo, da poznamo netrivialen razcep števila n:

$$n = a \cdot b,$$

$a, b \neq 1$ in $a \geq b$. Iščemo naravni števili x in y, za kateri velja

$$\begin{aligned} a &= x + y, \\ b &= x - y. \end{aligned}$$

Izrazimo x in y. Kratek račun da:

$$\begin{aligned} x &= \frac{a+b}{2}, \\ y &= \frac{a-b}{2}. \end{aligned}$$

Ker je število n liho, sta tudi a in b lihi števili. Njuna vsota oziroma razlika je torej sodo število in izraza $\frac{a+b}{2}$ in $\frac{a-b}{2}$ sta vedno pozitivni celi števili, razen ko je $b = a$, oziroma $n = a^2$, in je izraz $\frac{a-b}{2}$ enak 0. Torej x in y vedno obstajata. □

Opomba 1 *Če je n praštevilo, dobimo trivialen razcep*

$$n = 1 \cdot n = \left(\frac{n+1}{2}\right)^2 - \left(\frac{n-1}{2}\right)^2.$$

Opomba 2 *V primeru, ko je n praštevilo, dobimo največji x, torej je s tem določena zgornja meja za x:*

$$x \leq \frac{n+1}{2}.$$

Opomba 3 *V primeru, ko je n popoln kvadrat, dobimo najmanjši $x = \sqrt{n}$. V splošnem je $\sqrt{n}$ spodnja meja za x.*

Na primeru si oglejmo, kako poiščemo razcep lihega naravnega števila z uporabo trditve 1.

Primer 1 *Število 143 zapišemo po trditvi 1 kot:*

$$143 = x^2 - y^2 = (x - y)(x + y).$$

Poiskati želimo števili x in y.

Ostanki pri deljenju poljubnega števila s 4 so števila od 0 do 3. Kvadrati naravnih števil pa so lahko kongruentni le 0 in 1 po modulu 4:

$$y \equiv 0, 1, 2, 3 \pmod 4 \qquad \Rightarrow \qquad y^2 \equiv 0, 1, 0, 1 \pmod 4 \tag{1}$$

To velja za vsak kvadrat naravnega števila, tako x^2 kot y^2. Uporabimo še enakost $143 \equiv 3 \pmod 4$ in izračunajmo:

$$\begin{aligned} x^2 &\equiv 143 + y^2 \pmod 4 \\ x^2 &\equiv 3 + 0, 1 \pmod 4 \\ x^2 &\equiv 3, 0 \pmod 4. \end{aligned}$$

Ob upoštevanju (1) nam ostane le $x^2 \equiv 0 \pmod 4$, torej je $x \equiv 0, 2 \pmod 4$. Podoben razmislek naredimo še za praštevila 3, 5 in 7.

$$143 \equiv 2 \pmod 3, \quad 143 \equiv 3 \pmod 5, \quad 143 \equiv 3 \pmod 7;$$

$$\begin{aligned} y &\equiv 0, 1, 2 \pmod 3 && \Rightarrow && y^2 \equiv 0, 1, 1 \pmod 3 \\ y &\equiv 0, 1, 2, 3, 4 \pmod 5 && \Rightarrow && y^2 \equiv 0, 1, 4, 4, 1 \pmod 5 \\ y &\equiv 0, 1, 2, 3, 4, 5, 6 \pmod 7 && \Rightarrow && y^2 \equiv 0, 1, 4, 2, 2, 4, 1 \pmod 7. \end{aligned}$$

Tako dobimo rezultate:

$$\begin{aligned} y^2 &\equiv 0, 1 \pmod 3 \\ y^2 &\equiv 0, 1, 4 \pmod 5 \\ y^2 &\equiv 0, 1, 2, 4 \pmod 7. \end{aligned}$$

Izračunajmo pripadajoče kongruence za x^2 po modulu 5 in 7:

$$\begin{aligned} x^2 &\equiv 143 + y^2 \pmod 3 && \Rightarrow && x^2 \equiv 2, 0 \pmod 3 \\ x^2 &\equiv 143 + 0, 1, 4 \pmod 5 && \Rightarrow && x^2 \equiv 3, 4, 2 \pmod 5, \\ x^2 &\equiv 143 + 0, 1, 2, 4 \pmod 7 && \Rightarrow && x^2 \equiv 3, 4, 5, 6 \pmod 7. \end{aligned}$$

Izločimo enačbe, ki za kvadrate naravnih števil ne veljajo in dobimo vse dopustne ostanke števila x po modulih 3, 5 in 7.

$$\begin{aligned} x^2 &\equiv 0 \pmod 3 && \Rightarrow && x \equiv 0 \pmod 3 \\ x^2 &\equiv 4 \pmod 5 && \Rightarrow && x \equiv 2, 3 \pmod 5 \\ x^2 &\equiv 4 \pmod 7 && \Rightarrow && x \equiv 2, 5 \pmod 7 \end{aligned}$$

Strnimo naše ugotovitve. Iščemo tako število x za katerega velja $\sqrt{143} = 11.950... < x < \frac{143+1}{2} = 72$ in zadošča naslednjemu sistemu kongruenčnih enačb:

$$\begin{aligned} x &\equiv 0, 2 \pmod 4 \\ x &\equiv 0 \pmod 3 \\ x &\equiv 2, 3 \pmod 5 \\ x &\equiv 2, 5 \pmod 7 . \end{aligned}$$

Narišimo tabelo preveljanja modulskih enačb, iz katere razberemo najmanjše število x, ki ustreza vsem zahtevam. Čeprav pričakujemo, da je rešitev večja od $\sqrt{143}$, zaradi nazornosti pričnimo pri nič.

x	0	1	2	3	4	5	6	7	8	9	10	11	**12**	13	14	15	16
$x \pmod 4$	+	−	+	−	+	−	+	−	+	−	+	−	+	−	+	−	+
$x \pmod 3$	+	−	−	+	−	−	+	−	−	+	−	−	+	−	−	+	−
$x \pmod 5$	−	−	+	+	−	−	−	+	+	−	−	−	+	+	−	−	−
$x \pmod 7$	−	−	+	−	−	+	−	−	−	+	−	−	+	−	−	−	+

Najmanjša rešitev je $x = 12$. Ustreznemu x nato določimo pripadajočo vrednost y po enačbi $y^2 = x^2 - 143$. Razcep je torej oblike:

$$143 = 12^2 - 1^2 = 13 \cdot 11.$$

Iz tabele razberemo, da se v posamezni vrstici ciklično pojavlja vzorec. Ciklično ponavljanje je bistveno za izdelavo mehanskega stroja.

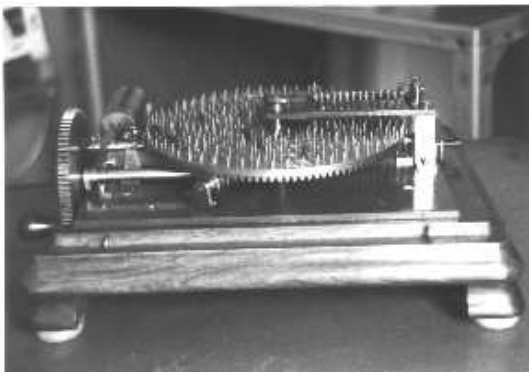
Razcep si oglejmo splošneje. Naštejmo vse količine, ki nastopajo v postopku:

- meji A in B za število x, $A < x < B$ (npr. $1 < x < 72$);
- množica modulov $m_1, m_2, ..., m_k$ $(k > 1)$ in $m_i > 1$, $i = 1, 2, ..., k$, ki so paroma tuja števila (npr. 3,4,5,7);
- k množic s sprejemljivimi ostanki: $R_i = \{r_{ij}; 0 \leq r_{ij} < m_i\}$, za $i = 1, 2, ..., k$ (npr. $R_3 = \{0\}$, $R_4 = \{0, 2\}$, $R_5 = \{2, 3\}$ in $R_7 = \{2, 5\}$).

Določiti moramo vsa naravna števila x, ki pri deljenju z modulom m_i dajo ostanke iz množice R_i, za vse $i = 1, 2, ..., k$. Oglejmo si osnovno idejo za izdelavo mehanskega stroja. Stroj, ki bi izvajal take operacije, potrebuje za vsak modul m_i krog ali obroč, razdeljen na m_i delov. Deli, ki predstavljajo sprejemljivo rešitev morajo biti na nek način označeni. Lego obročev preverjamo na določenem delu stroja, ki se imenuje *preiskovalna črta*. Obroči se usklajeno vrtijo. Rešitev dosežemo v trenutku, ko so vsi deli krogov znotraj okna označeni. *Števec* zabeleži število premikov, ki jih je izvedel stroj in iz njega preberemo rešitev. Vsako napravo takega tipa imenujemo *številsko rešeto* ali kratko kar *rešeto*.

3 Sestava in delovanje stroja

Leta 1913-1914 je začel E. O. Carissan, poročnik francoske pehote, razvijati svoje drugo številsko sito, toda delo je moral opustiti do leta 1919 zaradi II. svetovne vojne. Končen izdelek, imenovan '*machine à congruence*' ali slovensko '*kongruenčni stroj*', je dimenzij 27 cm × 33 cm × 12 cm.



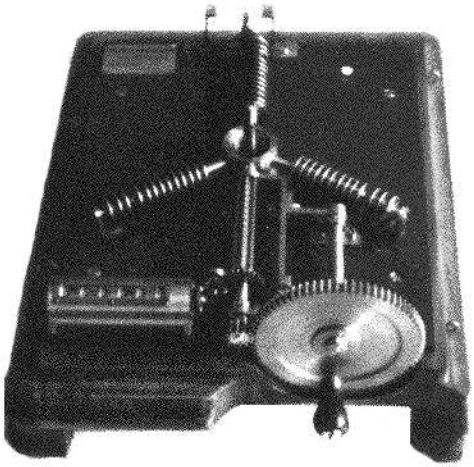
Slika 1: 'Kongruenčni stroj'

Oglejmo si podrobno sestavo stroja. Vsebuje 14 koncentričnih obročev iz medenine, vsak od njih pa predstavlja en modul m. Obroč, ki ustreza modulu m, ima na spodnji strani 2-milimetrske zobce, na zgornji strani pa m enako razmaknjenih železnih žebljičev, ki so pri vrhu okrepljeni. Moduli so fiksni: 19, 21, 23, 26, 29, 31, 34, 37, 41, 43, 47, 53, 55 in 59. Rešitev problema se nahaja v množici ostankov po modulu

$$\prod_{\substack{2\leq p\leq 59\\ p\ praštevilo}} p = 19\,22760\,35015\,42126\,39070$$

(glej [6], str. 77). Žebljiči na vsakem obroču so oštevilčeni s števili 1 do m. Območje, ki nosi oznako m, je obarvano rdeče. Koncentrični kongruenčni obroči sestavljajo krog in se vrtijo v skupni ravnini. Pri gibanju jim pomagajo tri tirnice, ki med seboj oklepajo kot 120° in segajo od osi vrtenja do največjega obroča. Vsaka tirnica ima 14 zelo tankih obročkov, ki nosijo kongruenčne obroče in zmanjšujejo trenje in škripanje. Ta čvrsta konstrukcija, ki se nahaja pod ravnino obročev vzdolž polmera kroga obročev in je vzporedna ravnini rotacije. Naprava je ročno vrtljiva preko ročaja, več manjših in enega velikega zobatega kolesa. Ročaj prek zobatih koles vrti *pogonsko gred*, ki skrbi za obračanje kongruenčnih obročev preko zobcev pod kongruenčnimi obroči. Prav tako zobata kolesca povezujejo 6-številčen števec z napravo.

Rešeto deluje tako, da na vse žebljiče, ki določajo ustrezne ostanke za modul m, namestimo neprevodne kapice. To storimo za vse module. Kadar sito prične z delovanjem na poziciji N (število na števcu), so obroči postavljeni tako, da



Slika 2: Podkonstrukcija stroja brez kongruenčnih obročev in ploščice armaturke.

žebljiči, ki predstavljajo vrednost $N \pmod m$, ležijo v posebni legi - postavljeni so v radialni koloni na *preiskovalni črti*, preko katere je položena *ploščica armaturka*. Ta vsebuje 14 stikal za izvir napetosti v ravni vrsti, tako da je nad vsakim obročem ravno eno stikalo. Pokriti žebljiči so nekoliko daljši in pritisnejo na stikalo, v trenutku ko gredo pod ploščico armaturko. Stikala na ploščici so zaporedno povezani, zato se v primeru, ko so na preiskovalni črti sami dovoljeni ostanki, sklene električen krog in to sproži zvok, podoben telefonskemu zvonenju. Zvok se sliši v slušalki operaterja, ki vrti ročico. V trenutku, ko zasliši zvok, vrtenje ustavi, ter počasi zavrti v nasprotno smer, dokler ponovno ne zasliši zvoka. Tedaj odčita število na števcu, ki je rešitev sistema kongruenčnih enačb.

Eugène Oliver Carissan je načrtoval nadgradnjo svojega rešeta, vendar je ni nikoli izvedel. Ročno vrtenje naj bi zamenjal elektromotor, ki bi bil v povezavi z armaturno ploščo tako, da bi se v primeru rešitve samodejno ustavil.

4 Testiranja in simulacije

Številsko rešeto E. O. Carissana je bilo prvič predstavljeno na razstavi *Exposition Publique de Machines à Calculer* v Parizu od 5. do 13. junija leta 1920. Stroj je bil preizkušen z nekaj nalogami. Za število 708 158 977 je v 10 minutah dokazal, da je praštevilo. Pokazal je, da ga lahko zapišemo le na en način kot vsoto kvadratov dveh števil

$$708158977 = 19224^2 + 18401^2,$$

kar je potrebni in zadostni pogoj za praštevilo (glej [2], [1]). Več primerov, ki jih je stroj rešil, najdemo v članku [1]. Poudariti moramo, da stroj ni iskal razcepe ali dokazoval nerazcepnost, ampak je reševal le sistem kongruenčnih enačb.

Zanimiva se zdi Carissanova izbira modulov. Še enkrat jih naštejmo:

$$19, 21, 23, 26, 29, 31, 34, 37, 41, 43, 47, 53, 55, 59.$$

Ni začel z najmanjšimi praštevili 2, 3, 5,...., temveč šele z 19; poleg tega vsi moduli niso praštevilski, števila pa med seboj niso paroma tuja. Najprej moramo razumeti, da naj bi stroji olajšali razcep velikih števil, tako je bilo smiselno namesto modulov 3 in 7 vzeti kar produkt $3 \cdot 7 = 21$. Isto velja za število $55 = 5 \cdot 11$. Toda razloge za izbiro dveh sodih števil 26 in 34 je potrebno iskati popolnoma drugje – v sami konstrukciji stroja. Obroči, ki predstavljajo module ležijo koncentrično. Najmanjši obroč predstavlja modul 19, največji pa modul 59. Vsak obroč je razdeljen na toliko enakih delov, kakršen modul predstavlja. V trenutku, ko je stroj pripravljen za delo so v preiskovalni črti poravnani ostanki števila N s števca po ustreznem modulu:

$$N \,(\mathrm{mod}\, 19)\,, N \,(\mathrm{mod}\, 21)\,, N \,(\mathrm{mod}\, 23)\,, ..., N \,(\mathrm{mod}\, 59)\,.$$

Ob takem zasuku ročice, ko se števec poveča za ena, pa preiskovalna črta kaže pozicijo

$$N + 1 \,(\mathrm{mod}\, 19)\,, N + 1 \,(\mathrm{mod}\, 21)\,, N + 1 \,(\mathrm{mod}\, 23)\,, ..., N + 1 \,(\mathrm{mod}\, 59)\,.$$

Vsak od kongruenčnih obročev je razdeljen na drugo število delov, njihovo gibanje pa je usklajeno tako, da se vsak premakne za enako število delcev na enkrat. Da si bomo lažje razumeli način gibanja obročev si poglejmo kolikšen kot zasuka napravi vsak izmed obročev posebej, kadar se števec poveča za 1:

$$\frac{360^\circ}{19}, \frac{360^\circ}{21}, \frac{360^\circ}{23}, ..., \frac{360^\circ}{59}.$$

Vsak obroč se torej zasuče za drugačen kot. Pogonska gred in zobci pod kongruenčnimi obročimorajo biti usklajeni tako, da se vsak kongruenčni obroč vrti s svojo kotno hitrostjo. Ob povečanju števila na števcu za 1, se pripadajoči žebljički na preiskovalni črti premaknejo na sosednjega, ki ima vrednost večjo za 1, razen v primeru, ko za m-tim žebljičem na obroču, ki predstavlja modul m, sledi žebljič 1 in ne $m + 1$ (mehanska simulacija prištevanja po modulu m). Tako natančna uskladitev delovanja obročev pa je bila lažje izvedljiva s skrbno izbranimi moduli, ki med seboj nimajo preveč velikega razpona. Števila 26, 34, 21 in 55 so zmanjšala prevelike vrzeli.

4.1 Računalniška simulacija stroja

V programskem paketu Matlab smo simulirali delovanje kongruenčnega stroja. Za module smo postavili natanko ista števila kot Carissan. Zaradi testiranja programa smo mu dodali funkcijo, ki nam nastavi sistem kongruenčnih enačb oziroma izračuna dopustne ostanke po danih modulih.

Računalniški program **Kongruencni_stroj** simulira delovanje Carissanovega stroja. Kot vhodna parametra podamo: **n** – število, ki ga cepimo in **N** – vrednost naravnega števila, za katerega začnemo preverjati kongruenčne enačbe in ustreza stanju na števcu Carissanovega stroja v trenutku, ko ga poženemo. Za razliko od mehanskega stroja dopustne ostanke po modulih ne vnašamo ročno (niso vhodni podatki), temveč jih pripravimo s posebno funkcijo **modulske_enacbe**, ki jo opišemo kasneje. Preiskovalni črti v programu ustreza vektor **ostanki**, kjer so zapisani ostanki trenutnega števila, ki ga prevejamo (števca), po Carissanovih modulih. Vrtenju ročice na stroju ustreza zanka po naravnih številih, ki se začne pri **N** in gre najdlje do $\frac{n+1}{2}$. V vsakem koraku zanke opravimo logično preverjanje kongruenčnih enačb. Pri mehanskem stroju to opravi armaturna ploščica s stikali. Če so vse enačbe izpolnjene namesto zvočnega signala prekinemo izvajanje programa in izpišemo poleg x, zaradi preverjanja rezultata, še faktorja razcepa ter njun produkt.

```
function [x,a,b,c]=Kongruencni_stroj(n,N)
m=[19;21;23;26;29;31;34;37;41;43;47;53;55;59];
for i=1:14
    dopustni_ostanki{i}=modulske_enacbe(n,m(i));
end
ostanki=mod(N,m);
for z=N:(n+1)/2
    for i=1:14
    kontrola(i)=any( ostanki(i)==dopustni_ostanki{i} );.
    %Kontrola je ena, če je i-ti ostanek med dopustnimi, sicer je nič
end
if all(kontrola)
    x=z;
    y=sqrt(x*x-n);
    a=x+y;
    b=x-y;
    c=a*b;
    break %Če smo našli rešitev prenehamo poganjanje stroja.
else
    ostanki=mod(ostanki+1,m);
    end
end
```

Pomožna funkcija **modulske_enacbe** pripravlja sistem modulskih enačb oziroma dopustne ostanke za dane module na način, prikazan v primeru 1. Opozorimo, da lahko stroj rešuje tudi modulske enačbe, pripravljene po drugih postopkih. V funkciji **modulske_enacbe** je v algoritmično obliko preveden postopek priprave enačb, ki temelji na razcepu po trditvi 1. Vhodna podatka sta: **n** – število, ki ga cepimo in **m** – modul, za katerega iščemo dopustne ostanke. Vse ostanke po modulu **m** kvadriramo in izračunamo ostanke kvadratov po modulu **m**. Tako pripravimo vektor **v2** ostankov kvadratov naravnih števil po danem modulu. Izmed ostankov kvadratov izberemo le dopustne, to je le tiste, ki zadoščajo kongruenčni enačbi

$$x^2 \equiv n + y^2 \pmod{m}\,, \tag{2}$$

ki sledi neposredno iz trditve 1. Vsi tisti ostanki pri deljenju po modulu m, katerih kvadrati zadoščajo enačbi (2), so izhodni podatek funkcije. Rezultate dobimo z uporabo pomožne funkcije **indeks_preseka**, ki dopustnim kvadratom poišče ustrezne ostanke za x.

```
function ostanki=modulske_enacbe(n,m)

o=mod(n,m);
v=0:m-1;
v2=mod(v.*v,m);
ostanki=[];
xkvadrat=mod(o+v2,m); %Enačbo x^2=n+y^2 gledamo po modulu m.
i=indeks_preseka(v2,unique(xkvadrat)); % x^2 mora ustrezati modulom,
                                         ki nastopajo v v2.
ostanki=i-1;
%Ker so elementi vektorja v urejeni (od 0 do m-1), je rezultat ravno "indeks elementa -1".


function IX = indeks_preseka(X,Y)
IX = [];
for i=1:max(size(Y))
    j = find(Y(i) == X); %V j shranimo vsa mesta v X, na katerih smo našli Y(i).
    if (~isempty(j))
        IX(end+1:end+length(j)) = j;
    end
end
```

4.2 Testiranje programa Kongruencni_stroj

Program **Kongruencni_stroj** smo testirali na nekaj manj zahtevnih primerih, saj je bil pomebnejši cilj preverjanje delovanja mehanskega stroja kot pa dejansko razcepljanje števil. Za testiranje smo uporabili osebni računalnik z vgrajenim procesorjem PIII s 1000Hhz in 256Mb spomina.

Poudariti moramo, da sistem modulskih enačb, ki ga dobimo s funkcijo **modulske_enacbe**, v splošnem ni enolično rešljiv. Samo kadar cepimo praštevilo imamo le en razcep. Poleg tega so rešitve odvisne od izbire modulov. Testiranje je pokazalo, da vrne funkcija **Kongruencni_stroj** poleg 'pravih' rešitev tudi napačne. Tako na primer pri številu 11111117 dobimo kar 33 'rešitev', ustrezna pa je le $x = 5555559$, ki predstavlja razcep $11111117 = 11111117 \cdot 1$. Pri ostalih pripadajoč razcep ni celoštevilski. Da bi poiskali razlog tej 'napaki', smo število razcepili še s sorodnim računalniškim programom, ki se je od predstavljenega ločil le po izbiri modulov. Tokrat smo za module izbrali vsa praštevila manjša od 60 in program ni našel nobene napačne rešitve več. Carissonov stroj je torej dal kakšno nepravo rešitev, zato je moral upravljalec stroja vsako rešitev ročno preveriti in po potrebi nadaljevati postopek iskanja. Sestavljeni moduli 21, 26, 34 in 55 so bili vir nepravih rešitev. Carisson je kljub temu, da je prihranil le tri obroče, izbral sestavljene module. Razlog je verjetno v težavni mehanski izvedbi gibanja kongruenčnih obročev, če bi bili moduli preveč različni.

Pomembna lastnost programa je njegova majhna prostorska zahtevnost; zato pa je časovna zahtevnost precej večja. V primerjavi z vgrajeno funkcijo factor je program precej počasnejši. Vendar vgrajena funkcija pri razcepu števila n le to deli z vsemi praštevilskimi faktorji do $\sqrt{n}$, tak algoritem pa pri razcepu velikih sestavljenih števil ni uporaben. Ob uporabi paralelnega procesiranja pa s predstavljeni algoritem brez težav razdelimo na več, med sabo povsem neodvisnih delov. To pomeni, da osrednjo zanko programa, kjer preverjamo kandidate za število x, razdelimo na več delov, vsak del pa obdela posamezen procesor (računalnik). Algoritem je primeren za razcepe števil, kjer sta faktorja približno enakih velikosti (glej [5], str. 200). Prav števila, ki imajo v razcepu le dva približno enako velika prafaktorja pa igrajo pomembno vlogo pri kriptografskih metodah. Kadar je n produkt dveh podobno velikih praštevil je y majhen in najmanjšo rešitev $x = \sqrt{n + y^2}$ dosežemo zelo hitro (po zelo malo korakih zunanje zanke), saj zanko pričnemo pri $\sqrt{n}$.

Učinkovitost programa je res izredna, če je razlika med praštevilskima faktorjema majhna. Za ilustracijo smo razcepili število

$$123\,456\,943\,209\,923 = 11\,111\,117 \cdot 11\,111\,119.$$

Že v prvem koraku je $x = 11\,111\,118$ zadostil vsem kongruenčnim enačbam in vrnil pravilen razcep. Potrebovali smo 4 889 operacij v pomični piki in 0.03s. Vgrajena funkcija factor je potrebovala 2 200 708 operacij in 25s.Pri večji razliki med faktorjema program v svoji preprosti obliki ni več tako zelo učinkovit. Za

razcep števila

$$288\,066\,008\,230\,459 = 11\,111\,117 \cdot 25\,925\,927$$

smo potrebovali 86 582 092 operacij v pomični piki, vgrajena funkcija factor pa le 3 269 539 operacij. Vendar pa predstavljeni program dobi rešitev v vsakem primeru, medtem ko za vgrajeno funkcijo to ne velja. Na primerjalnem računalniku z manj (128Mb) delovnega spomina vgrajena funkcija zaradi visoke prostorske zahtevnosti ni dobila rešitve v realnem času. Če v program za račun ostankov namesto deljenja po modulu (vrstica ostanki=mod(ostanki+1,m);) uporabimo krožni seznam (analogija obročem z moduli), dobimo le še četrtino (21 649 249) operacij v pomični piki, za razcep pa smo potrebovali približno 16 minut.

Seveda lahko program še precej izboljšamo. Prvo možno izboljšavo programa smo že spoznali; to je uporaba zgolj praštevilskih modulov. Poleg tega pa bi lahko program še dodatno razširili. V funkciji **modulske_enacbe** smo uporabili zelo preprosto metodo za pridobivanje kongruenčnih enačb, saj je zadoščala za testiranje računalniške simulacije kongruenčnega stroja. Funkcijo **modulske_enacbe** lahko nadomestimo s strožjo, toda to da v splošnem več kongruenčnih enačb, kar pa pomeni tudi počasnejše delovanje programa. Poleg tega je kongruenčni stroj namenjem le razcepu lihih števil. Razširjen računalniški program lahko na začeteku preveri deljivost s številom dva, morda pa je smiselno tudi, da preveri deljivost z majnimi praštevili.

5 Zaključek

Iskanju edinega preostalega mehanskega rešeta je botrovalo navdušenje avtorjev članka [1]. Eden od njih (F. Morain) se tudi sam ukvarja z razcepom velikih števil. Številsko rešeto E. O. Carissana je matematično natančno načrtovan in tehnično dovršen stroj, ki bi zmogel faktorizirati tudi števila z 22-mestnim faktorjem, kar je bilo leta 1920 velik dosežek. Tudi danes na običajnem osebnem računalniku to ni enostavno doseči. Največje do sedaj razstavljeno običajno sestavljeno število pa ima 'le' 155 decimalnih mest.

Tudi ob vseh navedenih izboljšavah bi se predstavljeni računalniški program, le težko kosal s časovnimi rezultati Carissanovega mehanskega stroja (za primerjavo časovne zahtevnosti stroja smo se opirali na podatke iz [1]). Mehanski pristop se je izkazal za neprekosljivega že pri krožnem nanašanju ostankov po modulih, kar še kar učinkovito simuliramo z uporabo krožnih seznamov. Toda pri preverjanju izpolnitve vseh modulskih enačb opravimo preveč operacij, medtem ko mehanski stroj v trenutku sklene električni krog in z zvokom opozori operaterja.

Verjetno predstavljeni algoritem ne more konkurirati različnim sodobnim algoritmom, s katerimi dosegajo nove rekorde na področju razcepa velikih naravnih števil. Kljub temu pa je ideja, ki sega celo v 19. stoletje, in njena izvedba pod okriljem Carissana vredna občudovanja. Upajmo, da zaradi hitrega razvoja

in velike zmogljivosti sodobnih računalnikov mehanski pristopi ne bodo povsem utonili v pozabo. Vsekakor pa bodo računalniki, s vse bolj obsežnimi spomini, ohranili čudovite človeške iznajdbe, tako da nobena več ne bo popolnoma padla v pozabo.

Literatura

[1] J. Shallit, H. C. Williams, F. Morain, *Discovery of a Lost Factoring Machine,* The Mathematical Intelligencer, Vol. 17, No. 3, 1995, Springer-Verlag, New York, str. 41-47.

[2] I. Vidav, *Kako ugotovimo da je naravno število sestavljeno preden ga razstavimo?,* DMFA, Presek, 1995/96 str. 130-136.

[3] N. Koblitz, *A Course in Number Theory and Cryptography,* Springer-Verlag, New York, 1987.

[4] *Matlab,* The MathWorks, INC., Natick, 1999.

[5] H. LDavenport, *The Higher Arithmetic,* University Press, Cambridge, 1999

[6] T. Nagell, *Introduction to Number Theory,* John Wiley & Sons, INC., Neww York, 1951

[7] http://www.math.uwaterloo.ca/~shallit/Papers/carissan.html

[8] http://www.lix.polytechnique.fr/Labo/Francois.Morain/english-index.html

[9] http://www.utm.edu/research/primes/largest.html

[10] http://mathforum.com/dr.math/faq/faq.prime.num.html