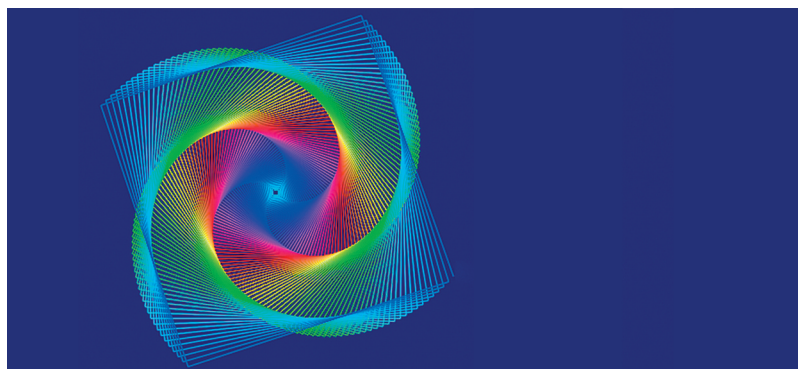


Geometrijski uzorci u skupu prirodnih brojeva

Vesna Skočir i
Franko Guberina, Zagreb



Povod za pisanje ovoga članka je neobičan algoritam Franka Guberine kojim se grafički mogu prikazati prirodni brojevi te njihovi podskupovi.

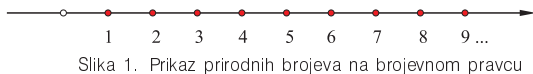
Toga smo se dana na satu informatike bavili kornjačinom grafikom u programskom jeziku Python. Kornjačina grafika je zanimljiva jer se rezultati programa mogu slikovno (grafički) prikazati na ekranu. Algoritam, odnosno crtež, učenici su trebali osmisлити sami. Kao kad dijete dobije prazan papir, bojice i – slobodu. No taj zadatak nisu dobili svi, već samo oni učenici koji su dobro savladali “temelje” kornjačine grafike. Ostali su rješavali jednostavne zadatke kako bi na taj način stekli potrebna znanja. Poruka je jasna: postoje određena temeljna znanja koja treba usvojiti kako bismo došli do rezultata te imali privilegiju izraziti svoje kreativne ideje i inovativnost. Ili, kao što pjeva Parni valjak: “moraš prvo puzati da bi znao hodati”.

U začetcima matematičke znanosti bilo je *brojenje*. Potreba za brojenjem stvari vjerojatno je jedan od prvih problema s kojim su se susreli stari narodi. Zapis broja te razvoj aritmetike dovest će do koncepta prirodnog broja bez kojega matematiku ne možemo ni zamisliti. Skup prirodnih brojeva te različiti podskupovi toga skupa bili su privlačni i onim najstarijim matematičarima. Njihova matematička terminologija nije poznavala pojam skupa, no to i nije bilo važno jer “priča” se može ispričati na različite načine.

Skup prirodnih brojeva uređen je po veličini. To znači da za svaka dva broja znamo koji je manji, a koji veći. “Nanižemo” li ove brojeve jednog za drugim tako da veći slijedi manjega, dobivamo niz prirodnih brojeva:

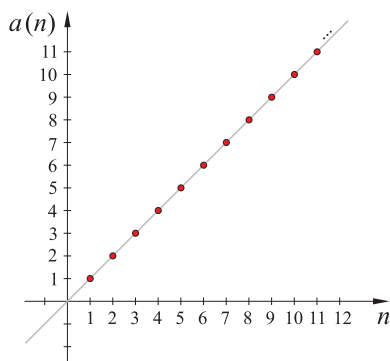
$$a : N \rightarrow N, \quad a_n = a(n) = n.$$

Možemo li zamisliti takav niz? Bit će jednostavnije predočiti ga grafički. Pogledajmo primjerice prikaz na brojnom pravcu (slika 1):



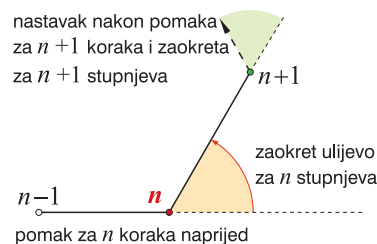
Budući da prirodnih brojeva ima beskonačno mnogo, moguće je prikazati samo dio toga niza. Na slici 1 prikazano je prvih devet članova niza. Vidimo da su točke “pravilno” raspoređene po pravcu i slutimo da će tako biti i dalje, tj. da neće biti gomilanja članova niza oko neke određene točke ma kako daleko odmakli na brojevnom pravcu.

Niz predstavlja funkciju pa ga možemo, kako je to uobičajeno za funkcije, prikazati u koordinatnom sustavu. Lijepo se vidi da točke niza leže na grafu linearne funkcije $f(x) = x$ te možemo naslutiti gdje će se nalaziti veliki brojevi toga niza (slika 2).



Slika 2. Prikaz niza prirodnih brojeva u koordinatnom sustavu

No je li to sve? Možemo li prirodne brojeve prikazati na neki nov i neuobičajen način? U ovom radu definirat ćemo algoritam s pomoću kojeg ćemo prirodne brojeve i njihove podskupove prikazati kao zanimljive geometrijske uzorke. Brojeve ćemo predstaviti točkama na zaslonu monitora s pomoću kornjačine grafike programskog jezika Python. Ideja je da prirodni broj n prikažemo na način da se od neke čvrste točke (prethodnika) pomaknemo za n jedinica prema naprijed, a potom zakrenemo za n stupnjeva ulijevo. Na slici 3 tako je prikazan broj n (crvena točka). Ta crvena točka sada postaje čvrsta točka za broj koji slijedi, a to je $n + 1$ (zeleni točka). Početna čvrsta točka smještena je u sredinu prozora.



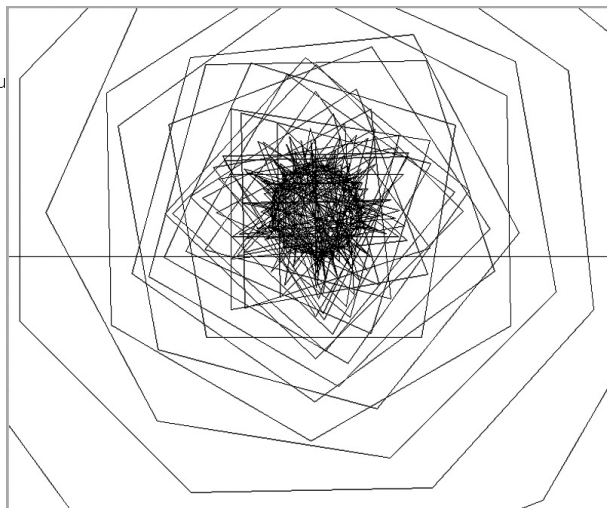
Slika 3. Crvenom i zelenom točkom prikazani su brojevi n i $n + 1$.

Prikažimo sada na opisani način prvih 50 000 prirodnih brojeva. Dovoljno je svega nekoliko redaka programskoga koda (program 1):

Program 1:

```
from turtle import*
def main():
    tracer(False)
    for i in range(1, 50000+1):
        fd(i)
        lt(i)
    return
main()
mainloop()
```

Rezultat je grafički uzorak prikazan na slici 4. No brojevi brzo rastu, pa su neke točke “pobjegle” izvan prozora.



Slika 4. Grafički uzorak generiran programom 1

Da bismo vidjeli cjelovitiji prikaz, dopunit ćemo naš program umanjujući sliku (program 2). Korisnik sada upisuje koliko puta želi smanjiti uzorak. Pritom

kut rotacije ostaje isti, a smanjuje se korak prema naprijed.

Program 2:

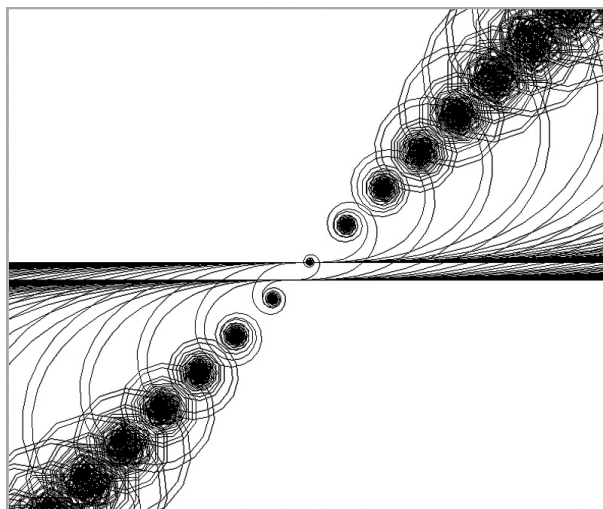
```
from turtle import*
def main():
    smanjenje = int(input('Koliko puta
    želite smanjiti prikaz: '))
    tracer(False)
    for i in range(1,50000+1):
        fd(i/smanjenje)
        lt(i)
    return
main()
mainloop()
```

Na slici 5 prikazan je isti skup brojeva smanjen 100 puta, no još uvijek nisu vidljivi svi brojevi. To se može ispraviti još većim smanjenjem. Primjerice, slika 6 smanjena je 1500 puta i to je dovoljno da cijeli uzorak s 50 000 prirodnih brojeva bude vidljiv u grafičkom prozoru.

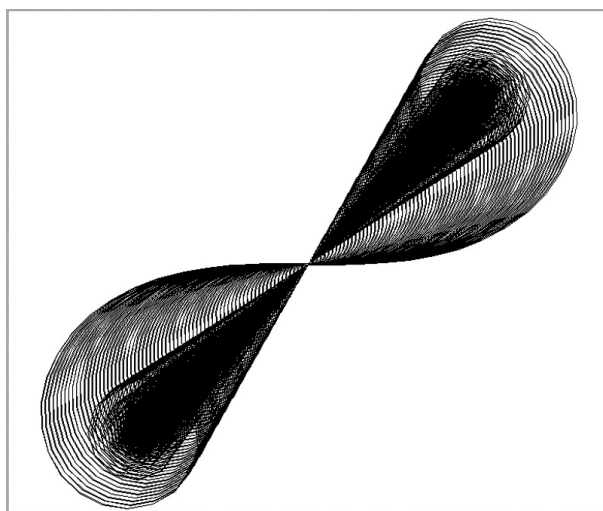
Bilo bi zanimljivo anketirati slučajne prolaznike i pitati ih bi li, u slučaju da pronađu ovakve slike, iste prosljedili matematičarima ili likovnim umjetnicima. No teško je vjerovati da bi ih povezali s prirodnim brojevima.

Zadržimo se još malo na skupu prirodnih brojeva. Već su stari matematičari među prirodnim brojevima izdvojili **proste** ili **primbrojeve**, to jest one koji su djeljivi samo s 1 i sa samim sobom (2, 3, 5, 7, 11, 13, 17, 23, 29...). Uočili su da takvih brojeva ima puno. Naslutili su da su prosti brojevi nepravilno raspršeni među svim prirodnim brojevima. No jesu li zaista? I kakva je raspodjela skupa prostih brojeva unutar skupa svih prirodnih brojeva? Trebalo je proći dosta godina (stoljeća) da se pronađe zakon raspodjele prostih brojeva. Za to su zaslužni matematičari Jacques-Salomon Hadamard (1865. – 1963.) i Charles Jean de la Vallée-Poussin (1866. – 1962.) koji su 1896. godine nezavisno jedan od drugoga uspjeli dokazati takvu raspodjelu. Ako s $\pi(x)$ označimo broj prostih brojeva manjih od ili jednakih broju x , tada vrijedi:

$$\lim_{x \rightarrow \infty} \frac{\pi(x)}{x / \ln x} = 1.$$



Slika 5. Grafički uzorak generiran programom 2 (slika 100 puta smanjena)



Slika 6. Grafički uzorak generiran programom 2 (slika 1500 puta smanjena)

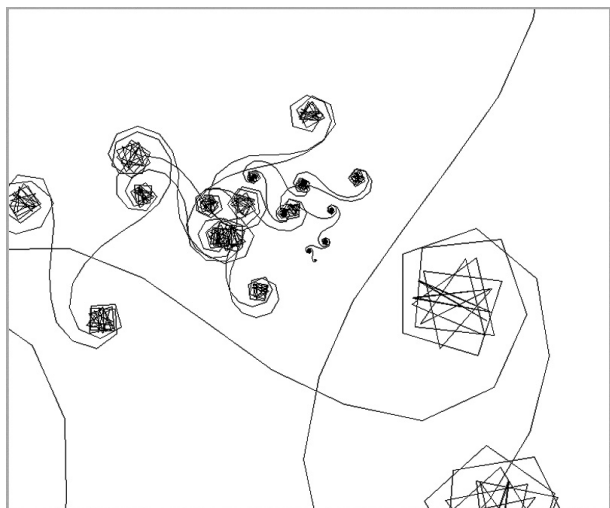
U ovom radu zadržat ćemo se na uzorcima koje u grafičkom prozoru "crtaju" prosti brojevi po istom principu kao i za prirodne brojeve. Pogledajmo kako to izgleda za prvih 100 000 prostih brojeva (program 3, slike 7, 8 i 9).

Program 3:

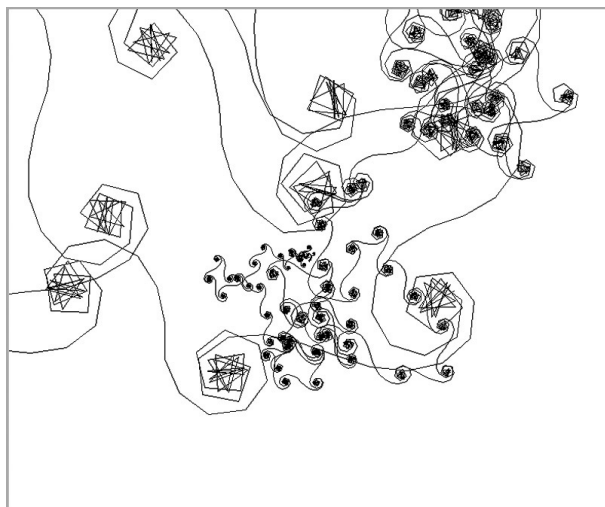
```
from turtle import*
def prost (n):
    import math
    from math import sqrt
    prost = True
    for i in range (2, round (sqrt(n)+1)):
        if n % i == 0:
            prost = False
            break
    if n != 1 and prost:
        prost = True
    else:
        prost = False
    return prost

def main ():
    n = 100000
    smanjenje = int (input('Koliko puta
    želite smanjiti prikaz: '))
    tracer(False)
    for i in range (1, n+1):
        if prost (i):
            fd(i/smanjenje)
            lt(i)
    return

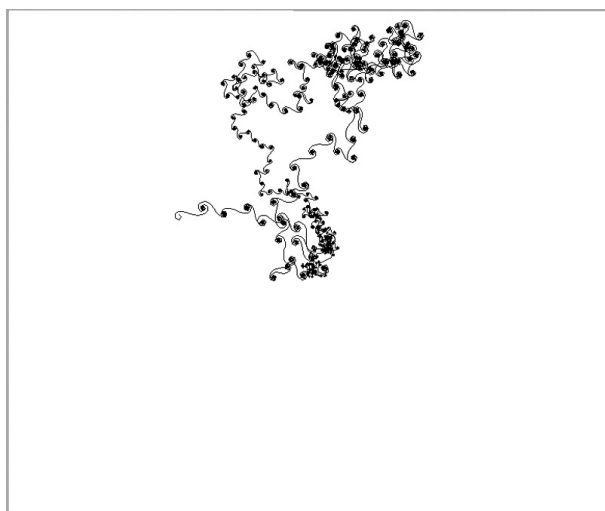
main ()
mainloop ()
```



Slika 7. Grafički uzorak generiran programom 3 (slika 200 puta smanjena)



Slika 8. Grafički uzorak generiran programom 3 (slika 2000 puta smanjena)



Slika 9. Grafički uzorak generiran programom 3 (slika 20000 puta smanjena)

Geometrijski motivi asociraju na teoriju determinističkog kaosa – veliku i popularnu teoriju 20. stoljeća. Kamo nas ovakav prikaz može dovesti? Teško je reći. U svakom slučaju, nadamo se da se slažete s nama, pruža nam još jedno iskustvo doživljaja matematike kao lijepe znanosti.