

PRESEK

List za mlade matematike, fizike, astronome in računalnikarje

ISSN 0351-6652

Letnik 11 (1983/1984)

Številka 4

Strani 161-163

Roman Rojko:

ZMAJEVA KRIVULJA

Ključne besede: matematika.

Elektronska verzija: <http://www.presek.si/11/672-Rojko.pdf>

© 1984 Društvo matematikov, fizikov in astronomov Slovenije

© 2010 DMFA - založništvo

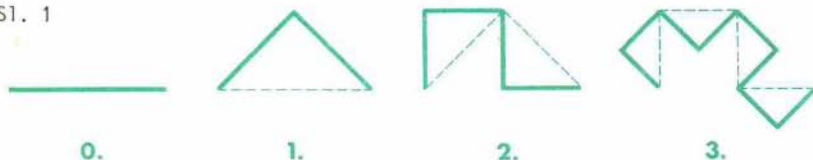
Vse pravice pridržane. Razmnoževanje ali reproduciranje celote ali posameznih delov brez poprejšnjega dovoljenja založnika ni dovoljeno.

ZMAJEVA KRIVULJA

Vzemimo v roke ozek papirnat trak in ga začnimo pregibati. Počnimo to tako: trak prepognimo čez polovico, nato še enkrat, pa še enkrat, nazadnje še n -tič. Potem pa te pregibe po vrsti odprimo, tako da bo trak tvoril v vsakem pregibu pravi kot. Trak (pravzaprav njegov tloris) je tako dobil obliko odprtega (nesklenjenega) pravokotnega mnogokotnika, ki nosi eksotično ime *zmajeve krivulja* (n -tega reda). Traku seveda ne moremo v nedogled pregibati, opisana *papirna metoda* zato hitro odpove.

Boljša je *geometrijska metoda*. Na papir narišimo daljico. Ta predstavlja zmajevo krivuljo reda 0 (ničkrat preganjen trak). Nad njo narišemo enkrat prelomljeno črto - zmajevo krivuljo prvega reda.

Sl. 1



Nad krivuljo 1. reda narišemo nato krivuljo 2. reda, in to takole: nad vsako daljico (stranico mnogokotnika) narišemo prelomljeno črto (kateti trikotnika), enkrat zgoraj, drugič spodaj. Na sploh lahko narišemo nad vsako zmajevo krivuljo njeno naslednico tako, da po vrsti lomimo stranice, izmenično na eno in na drugo stran.

Oglejmo si sedaj *dvojiško metodo* izdelovanja zmajevih krivulj. Vzemimo kako zmajevo krivuljo, ji stopimo na glavo in potujmo po njej proti repu. Sproti si zapisujemo pravokotne ovinke, 1 naj predstavlja levi, 0 pa desni ovinek. Tako lahko vsako zmajevo krivuljo zapišemo v obliki dvojiškega števila. Nas seveda zanima nasprotna smer: kako sestaviti zaporedje dvojiških števil, s pomočjo katerega bomo lahko potem risali krivulje. To naredimo takole:

- prvo število je 1 (krivulja 1. reda),
- vsako naslednje število dobimo iz prejšnjega takole:
 - prejšnjemu številu najprej dopišemo 1,
 - nato pa mu dopišemo še prejšnje število samo s spremenjenim srednjim bitom.

Prvih nekaj zmajevih krivulj lahko zdaj zapišemo takole:

```
1
110
1101100
110110011100100
...
```

Čas je že, da opazimo nekatere lastnosti zaporedja zmajevih krivulj. Vsaka zmajeva krivulja natančno pokriva svojo predhodnico (privzeli smo seveda, da imajo vse krivulje enako dolge stranice). Število kotov in dolžina krivulje pa z rastočim redom zelo hitro naraščata. Zmajeva krivulja n -tega reda ima namreč

$$k_n = 2 k_{n-1} + 1 \text{ oziroma } k_n = 2^n - 1$$

kotov, število stranic pa je očitno za eno večje od števila kotov:

$$s_n = k_n + 1$$

(Pri sklenjenih mnogokotnikih je število stranic enako številu kotov.) Zaradi omejenosti človeške potrpežljivosti bomo risanje zmajevih krivulj zaupali računalniku.

Program bo uporabljal tako imenovano "turtle" grafiko (želvjo grafiko), kar bi verjetno najustrezneje prevedli z izrazom *polarna grafika*. Računalniki namreč poznajo dva načina uporabe grafičnega orodja. V *pravokotni grafiki* rišemo daljice tako, da v ukazu navedemo pravokotni koordinati obeh krajišč, recimo:

```
line(x1,y1,x2,y2)
```

Se bolj običajna pa sta ukaza

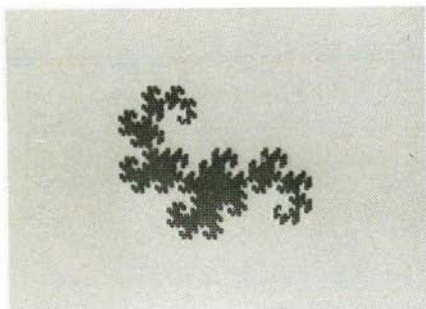
```
move(x,y)
```

za premik (dvignjenega) peresa na točko (x,y) in

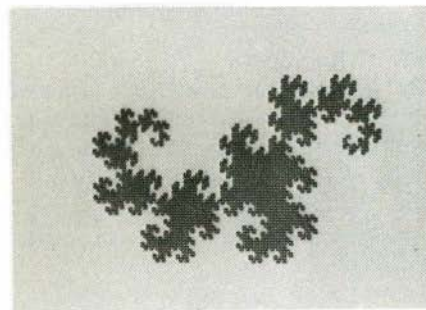
```
draw(x,y)
```

za risanje daljice od prejšnjega mesta peresa do točke (x,y) . Na grafičnem zaslonu imamo namesto peresa "žarek".

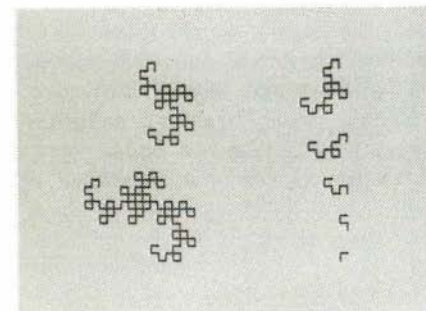
Polarna grafika pa uporablja namišljen risalni stroj (želvo), ki potuje po papirju, se obrača in riše. Krmilimo ga tako, da mu posredujemo polarne ko-



Sl. 2



Sl. 3



Sl. 4

ordinate (kot in razdaljo). Značilna ukaza sta

`turn(a),`

kar pomeni: obrni se za kot a , in

`draw(d),`

kar pa pomeni: nariši daljico z dolžino d enot v smeri, kamor si trenutno obrnjen.

Zdaj pa že lahko zapišemo podprogram za risanje zmajevih krivulj:

```

PROCEDURE zmaj (red: INTEGER);
BEGIN
  IF red = 0 THEN draw(d) ELSE
  IF red > 0
  THEN
    BEGIN
      zmaj(red - 1);
      turn(90);
      zmaj(-red + 1)
    END
  ELSE
    BEGIN
      zmaj(-red - 1);
      turn(-90);
      zmaj(red + 1)
    END
  END;

```

Na začetku moramo izbrati začetek krivulje, dolžino njene stranice d in sam red krivulje. Na slikah 2., 3 in 4 si lahko ogledamo nekaj računalnikovih izdelkov.

Roman Rojko