Számítógépes számelmélet

Járai Antal

Ezek a programok csak szemléltetésre szolgálnak

- 1. A prímek eloszlása, szitálás
- 2. Egyszerű faktorizálási módszerek
- 3. Egyszerű prímtesztelési módszerek
- 4. Lucas–sorozatok
- 5. Alkalmazások
- 6. Számok és polinomok
- 7. Gyors Fourier–transzformáció
- 8. Elliptikus függvények
- 9. Számolás elliptikus görbéken
- 10. Faktorizálás elliptikus gürbékkel
- 11. Prímteszt elliptikus görbékkel
- 12. Polinomfaktorizálás
- 13. Az AKS–teszt
- 14. A szita módszerek alapjai
- 15. Számtest szita

16. Vegyes problémák

```
> restart; with(numtheory);
```

[*GIgcd*, *bigomega*, *cfrac*, *cfracpol*, *cyclotomic*, *divisors*, *factorEQ*, *factorset*, *fermat*, *imagunit*, *index*, *integral_basis*, *invcfrac*, *invphi*, *issqrfree*, *jacobi*, *kronecker*, λ, *legendre*, *mcombine*, *mersenne*, *migcdex*, *minkowski*, *mipolys*, *mlog*, *mobius*, *mroot*, *msqrt*, *nearestp*, *nthconver*, *nthdenom*, *nthnumer*, *nthpow*, *order*, *pdexpand*, ϕ, π, *pprimroot*, *primroot*, *quadres*, *rootsunity*, *safeprime*, σ, *sq2factor*, *sum2sqr*, τ, *thue*] (16.1)

16.1. Collatz-probléma.

```
> interface(echo=3);
```

1 (16.1.1)

```
> ;
> #
  # The 3*x+1 problem of Collatz
  #

> L:=[2,{3}];
```

$L := [2, \{3\}]$ (16.1.2)

```
> points:=[[2,nops(L[2])]];
```

$points := [[2, 1]]$ (16.1.3)

```
> #
  # This tests an odd x wether in n halfing steps became
  smaller then x
  #

  test:=proc(n::posint,x::posint) local m,y;
  y:=x;
  m:=n;
  while m>0 do
    y:=3*y+1;
    while m>0 and type(y,even) do m:=m-1; y:=y/2 od;
    if y<x then RETURN(true) fi
  od; false end;
```

$test := \mathbf{proc}(n::posint, x::posint)$ (16.1.4)

$\mathbf{local}\ m, y;$

$y := x;$

$m := n;$

```
    while 0 < m do
        y := 3 * y + 1;
        while 0 < m and type(y, even) do
            m := m - 1;
            y := 1 / 2 * y
        end do;
        if y < x then
            RETURN(true)
        end if
    end do;
    false
end proc
```

```
> reduce:=proc(n,S) local x,SS;
  SS:={};
  for x in S do if not test(n,x) then SS:=SS union {x} fi od;
  SS end;
```

```
reduce := proc(n, S)                                        (16.1.5)
    local x, SS;
    SS := {};
    for x in S do
        if not test(n, x) then
            SS := union(SS, {x})
        end if
    end do;
    SS
end proc
```

```
> nextL:=proc(L) local n,S,x,y;
  n:=L[1]; S:=L[2]; S:=S union map(proc(x,y) x+2^y end,S,n);
  [n+1,reduce(n+1,S)] end;
```

```
nextL := proc(L)                                            (16.1.6)
    local n, S, x, y;
    n := L[1];
    S := L[2];
    S := union(S, map(proc(x, y)
        x + 2 ^ y
```

```
    end proc, S, n));
    [n + 1, reduce(n + 1, S)]
end proc
```

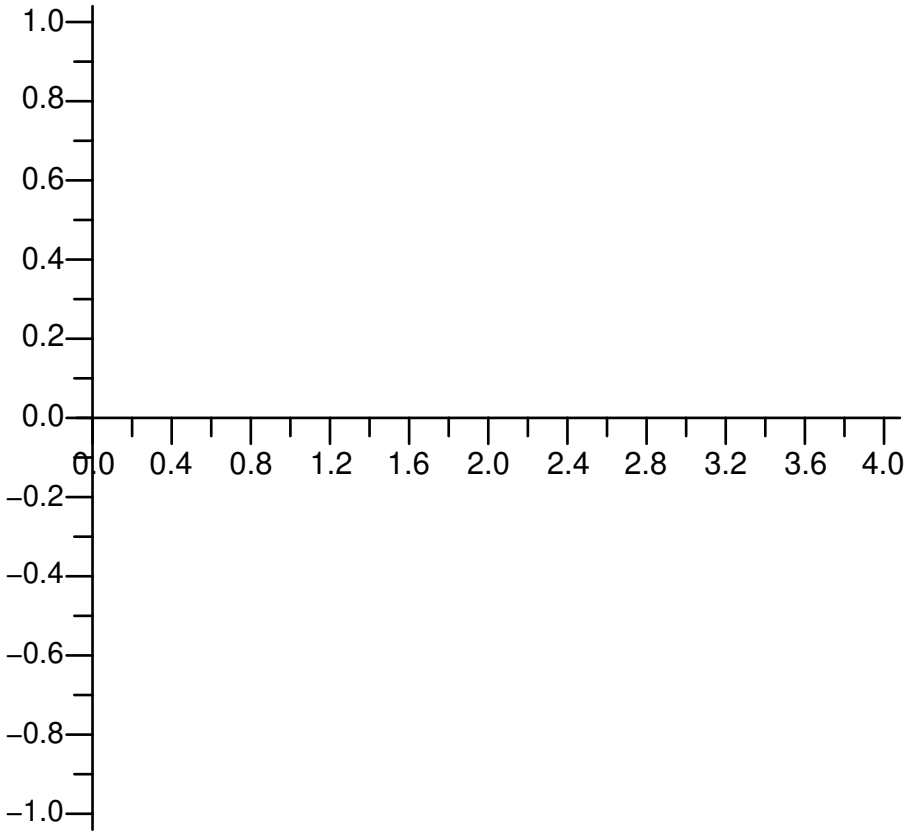
```
> cycle:=proc(n::posint) local L,points;
  L:=[2,{3}];
  points:=[[2,nops(L[2])]];
  while L[1]<n do
    L:=nextL(L);
    points:=[op(points),[L[1],nops(L[2])]];
  od end;
```

$$cycle := \mathbf{proc}(n::posint) \tag{16.1.7}$$

```
    local L, points;
    L := [2, {3}];
    points := [[2, nops(L[2])]];
    while L[1] < n do
        L := nextL(L);
        points := [op(points), [L[1], nops(L[2])]]
    end do
end proc
```

```
> logpoints:=map(`x`->[x[1],evalf(ln(x[2]))],points); plot
  (logpoints);
```

$$logpoints := [[2, 0.]]$$



16.2. Iteration of the lambda function.

```
> #
  # Definition of the lambda function
  #

  la:= proc(x) sigma(x)-x end;
```

$$la := \mathbf{proc}(x)\ numtheory:-\sigma(x) - x\ \mathbf{end\ proc} \tag{16.2.1}$$

```
> lacycle:=proc(x::posint) local y,z;
  z:=x; y:=la(x); print(y);
  while not y=z do
    y:=la(y); print(y);
    y:=la(y); print(y);
    z:=la(z); od;
  end;
```

$$lacycle := \mathbf{proc}(x::posint) \tag{16.2.2}$$

$$\mathbf{local}\ y, z;$$

```
    z := x;
    y := la(x);
    print(y);
    while not y = z do
        y := la(y);
        print(y);
        y := la(y);
        print(y);
        z := la(z)
    end do
end proc
> lacycle(16777778);
                                8924494
                                4530794
                                2361334
                                1418666
                                 716758
                                 511994
                                 365734
                                 182870
                                 146314
                                 109160
                                 136540
                                 150236
                                 128476
                                  96364
                                  72280
                                 104120
                                 144280
                                 180440
                                 258040
                                 322640
                                 454840
```

588440
768040
1368920
2151880
2902520
3685480
4666520
5833240
9407720
14784280
26050520
36330280
62021720
83821480
105128120
136915000
187824800
270704446
138273914
76289146
38195354
20495194
10275194
5614438
3173450
3573142
2064458
1349206
674606
354634
263414
131710

105386
67414
36554
27400
36770
29434
14720
22000
36032
35596
32444
24340
26816
26524
22476
29996
22504
21596
16204
12160
18440
23140
29780
32800
49226
25558
15770
14470
11594
9142
6554
3706

2234
1120
1904
2560
3578
1792
2296
2744
3256
3584
4600
6560
9316
8072
7078
3542
3370
2714
1606
1058
601
1
0
0
0
0
0
0
0
0
0

0

0

0

0

0 (16.2.3)

16.3. Blum–rejtjelzés.

>

16.2. Lánctörtek.

>

16.3. Kvadratikus irracionális számok lánctört alakja.

16.4. Faktorizálás lánctörtekkel.

>

16.5. Négyzetes szita.

>

16.6. Többpolinomos négyzetes szita.

>

16.7. letlen négyzet módszere.

>