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

```
> restart;
```

▶ 7.1. Polinomszorzás gyors Fourier–transzformációval.

▼ 7.2. Gyors Fourier–transzformáció (FFT).

```
>
```

▼ 7.3. Inverz FFT.

```
>
```

▼ 7.4. Szorzás komplex FFT–vel.

```
>
```

▼ 7.5. Valós FFT.

```
>
```

7.6. Szorzás komplex FFT-vel a gyakorlatban.

```
> with(plots);
```

[*Interactive*, *animate*, *animate3d*, *animatecurve*, *arrow*, *changecoords*, *complexplot*, *complexplot3d*, *conformal*, *conformal3d*, *contourplot*, *contourplot3d*, *coordplot*, *coordplot3d*, *cylinderplot*, *densityplot*, *display*, *display3d*, *fieldplot*, *fieldplot3d*, *gradplot*, *gradplot3d*, *graphplot3d*, *implicitplot*, *implicitplot3d*, *inequal*, *interactive*, *interactiveparams*, *listcontplot*, *listcontplot3d*, *listdensityplot*, *listplot*, *listplot3d*, *loglogplot*, *logplot*, *matrixplot*, *multiple*, *odeplot*, *pareto*, *plotcompare*, *pointplot*, *pointplot3d*, *polarplot*, *polygonplot*, *polygonplot3d*, *polyhedra_supported*, *polyhedraplot*, *replot*, *rootlocus*, *semilogplot*, *setoptions*, *setoptions3d*, *spacecurve*, *sparsematrixplot*, *sphereplot*, *surfdata*, *textplot*, *textplot3d*, *tubeplot*] (7.6.1)

```
> L:=[[1972,130],[1982,400],[1985,800],[1985,1700],[1988,2670],
  [1991,16000],[1995,100000]];
  LL:=map(x->[x[1],log[10.](x[2])],L);
  L1:=[[1982,0.4],[1996,400]];
  LL1:=map(x->[x[1],log[10.](x[2])],L1);
  fti:=[TIMES,BOLD,25]; f1:=[TIMES,BOLD,15]; fa:=[TIMES,BOLD,
  15];
```

$$L := [[1972, 130], [1982, 400], [1985, 800], [1985, 1700], [1988, 2670], [1991, 16000], [1995, 100000]]$$

$$LL := [[1972, 2.113943352], [1982, 2.602059991], [1985, 2.903089987], [1985, 3.230448921], [1988, 3.426511261], [1991, 4.204119983], [1995, 5.000000000]]$$

$$L1 := [[1982, 0.4], [1996, 400]]$$

$$LL1 := [[1982, -.3979400087], [1996, 2.602059991]]$$

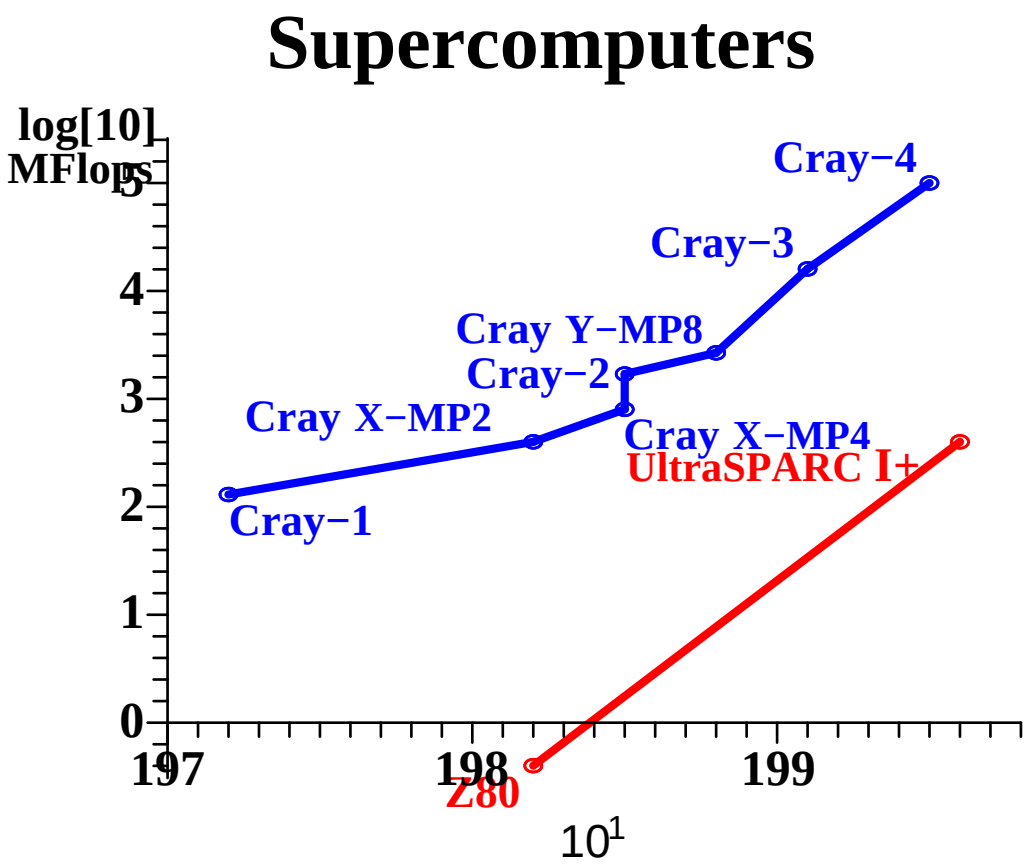
$$fti := [TIMES, BOLD, 25]$$

$$f1 := [TIMES, BOLD, 15]$$

$$fa := [TIMES, BOLD, 15] \quad (7.6.2)$$

```
> display([plot(LL,1970..1998,color=blue,titlefont=fti,title=
  'Supercomputers',
  thickness=3),
  plot(LL,1970..1998,color=blue,style=POINT,symbol=CIRCLE),
  plot(LL1,1970..1998,color=red,thickness=3,axesfont=fa),
  plot(LL1,1970..1998,color=red,style=POINT,symbol=CIRCLE),
```

```
textplot([1970,5.3,`log[10] `],align={ABOVE,LEFT},font=fa),
textplot([1970,4.9,`MFlops `],align={ABOVE,LEFT},font=fa),
textplot([LL1[1][1],LL1[1][2],`Z80 `],align={BELOW,LEFT},
color=red,font=f1),textplot([LL1[2][1],LL1[2][2],`UltraSPARC
I+   `],align={BELOW,LEFT},color=red,font=f1),textplot([LL[1]
[1],LL[1][2],`Cray-1 `],align={BELOW,RIGHT},color=blue,font=
f1),
textplot([LL[2][1],LL[2][2],`Cray X-MP2   `],align={ABOVE,
LEFT},color=blue,font=f1),
textplot([LL[3][1],LL[3][2],`Cray X-MP4 `],align={BELOW,
RIGHT},color=blue,font=f1),
textplot([LL[4][1],LL[4][2],`Cray-2 `],align={LEFT},color=
blue,font=f1),
textplot([LL[5][1],LL[5][2],`Cray Y-MP8 `],align={ABOVE,LEFT}
,color=blue,font=f1),
textplot([LL[6][1],LL[6][2],`Cray-3 `],align={ABOVE,LEFT},
color=blue,font=f1),
textplot([LL[7][1],LL[7][2],`Cray-4 `],align={ABOVE,LEFT},
color=blue,font=f1)]);
```




```
> b:=2.; WS:=32;

  schmul:=[[5*WS,0.00003],[100*WS,0.007],[10382*WS,3.]];

  classicmul:=[[WS,10.0/(4*10^7)],[2^5*WS,10.0*2^10/(4*10^7)]];

  classicsqr:=[[WS,10.0/(4*10^7)],[2*WS,3*10.5/(4*10^7)],
  [4*WS,10*10.5/(4*10^7)],[8*WS,34*10.5/(4*10^7)],
  [16*WS,136*10.5/(4*10^7)],[2^5*WS,528*10.5/(4*10^7)]];

  karamul:=[[2^2*WS,16.14/(4*10^6)],[2^3*WS,49.5/(4*10^6)],
  [2^4*WS,16.74/(4*10^5)],[2^5*WS,54.25/(4*10^5)],[2^6*WS,
  17.17/(4*10^4)],
  [2^7*WS,53.19/(4*10^4)],[2^8*WS,16.3/(4*10^3)],[2^9*WS,50.08/
  (4*10^3)],
  [2^10*WS,15.3/400],[2^11*WS,46.78/400],[2^12*WS,14.25/40],
  [2^13*WS,43.09/40],[2^14*WS,129.88/40],[2^15*WS,390.79/40]];
```

```
sqr:=[[2^2*WS,10.62/(4*10^6)],[2^3*WS,34.84/(4*10^6)],
[2^4*WS,12.44/(4*10^5)],[2^5*WS,43.02/(4*10^5)],[2^6*WS,
14.32/(4*10^4)],
[2^7*WS,46.24/(4*10^4)],[2^8*WS,14.65/(4*10^3)],[2^9*WS,45.9/
(4*10^3)],
[2^10*WS,14.37/400],[2^11*WS,44.24/400],[2^12*WS,13.59/40],
[2^13*WS,41.44/40],[2^14*WS,125.95/40],[2^15*WS,381.62/40]];

karasqr:=[[2^2*WS,10.89/(4*10^6)],[2^3*WS,34.45/(4*10^6)],
[2^4*WS,11.78/(4*10^5)],[2^5*WS,37.77/(4*10^5)],[2^6*WS,
11.94/(4*10^4)],
[2^7*WS,36.49/(4*10^4)],[2^8*WS,10.61/(4*10^3)],[2^9*WS,
32.39/(4*10^3)],
[2^10*WS,97.29/(4*10^3)],[2^11*WS,29.67/400],[2^12*WS,
90.12/400],
[2^13*WS,26.61/(40*(32/33))],[2^14*WS,80.18/(40*(32/33))],
[2^15*WS,240.61/(40*(32/33))]];

fftmul:=[[7*WS*2,9.96/(4*10^4)],[7*WS*2^2,21.18/(4*10^4)],
[7*WS*2^3,45.44/(4*10^4)],[7*WS*2^4,9.79/(4*10^3)],
[7*WS*2^5,21.2/(4*10^3)],[7*WS*2^6,46.85/(4*10^3)],
[15*WS*2^5,51.53/(4*10^3)],[7*WS*2^7,10.42/400],
[15*WS*2^6,11.16/400],[7*WS*2^8,23.12/400],[15*WS*2^7,
24.15/400],
[7*WS*2^9,50.06/400],[15*WS*2^8,51.54/400],[7*WS*2^10,
10.96/40],
[15*WS*2^9,10.94/40],[15*WS*2^10,23.15/40],[15*WS*2^11,
49.31/40]];

fftsqrold:=[[7*WS*2,73.51/(4*10^5)],[7*WS*2^2,15.62/(4*10^4)]
,
[7*WS*2^3,33.21/(4*10^4)],[7*WS*2^4,70.96/(4*10^4)],
[7*WS*2^5,15.45/(4*10^3)],[7*WS*2^6,33.06/(4*10^3)],
[15*WS*2^5,38.68/(4*10^3)],[7*WS*2^7,72.59/(4*10^3)],
[15*WS*2^6,82.1/(4*10^3)],[7*WS*2^8,16.42/400],[15*WS*2^7,
18.05/400],
[7*WS*2^9,35.79/400],[15*WS*2^8,38.36/400],[7*WS*2^10,
79.67/400],
[15*WS*2^9,81.51/400],[15*WS*2^10,17.42/40],[15*WS*2^11,
37.43/40]];

fftsqr:=[[7*WS*2,70.43/(4*10^5)],[7*WS*2^2,14.92/(4*10^4)],
[7*WS*2^3,31.71/(4*10^4)],[7*WS*2^4,67.90/(4*10^4)],
[7*WS*2^5,14.60/(4*10^3)],[7*WS*2^6,31.32/(4*10^3)],
[15*WS*2^5,34.34/(4*10^3)],[7*WS*2^7,69.02/(4*10^3)],
[15*WS*2^6,73.35/(4*10^3)],[7*WS*2^8,15.64/400],[15*WS*2^7,
16.24/400],
[7*WS*2^9,33.90/400],[15*WS*2^8,34.71/400],[7*WS*2^10,
```

```
74.26/400],
[15*WS*2^9,73.64/400],[15*WS*2^10,16.03/40],[15*WS*2^11,
33.53/40]];

rfffftmulc:=[[22*2^5,14.80/(4*10^4)],[21*2^6,32.03/(4*10^4)],
[21*2^7,64.06/(4*10^4)],[20*2^8,14.34/(4*10^3)],
[20*2^9,30.21/(4*10^3)],[19*2^10,70.39/(4*10^3)],[19*2^11,
16.81/400],
[18*2^12,35.45/400],[17*2^13,74.06/400],[17*2^14,15.59/40],
[16*2^15,32.32/40],[16*2^16,67.46/40]];

rfffftmul2ss:=[[22*2^5,73.11/(4*10^5)],[21*2^6,14.35/(4*10^4)]
,
[21*2^7,28.78/(4*10^4)],[20*2^8,62.45/(4*10^4)],
[20*2^9,14.59/(4*10^3)],[19*2^10,35.65/(4*10^3)],[19*2^11,
92.16/(4*10^3)],
[18*2^12,18.36/400],[17*2^13,42.46/400],[17*2^14,85.89/400],
[16*2^15,18.37/40],[16*2^16,37.84/40]];

rfffftsqrc:=[[22*2^5,12.83/(4*10^4)],[21*2^6,26.39/(4*10^4)],
[21*2^7,54.81/(4*10^4)],[20*2^8,11.35/(4*10^3)],
[20*2^9,24.22/(4*10^3)],[19*2^10,54.75/(4*10^3)],[19*2^11,
13.53/400],
[18*2^12,28.36/400],[17*2^13,60.39/400],[17*2^14,12.04/40],
[16*2^15,25.62/40],[16*2^16,52.58/40]];

rfffftsqr2ss:=[[22*2^5,51.75/(4*10^5)],[21*2^6,10.44/(4*10^4)]
,
[21*2^7,19.83/(4*10^4)],[20*2^8,43.66/(4*10^4)],
[20*2^9,95.11/(4*10^4)],[19*2^10,25.53/(4*10^3)],[19*2^11,
63.42/(4*10^3)],
[18*2^12,13.22/400],[17*2^13,28.36/400],[17*2^14,59.23/400],
[16*2^15,12.56/40],[16*2^16,25.85/40]];
```

$$b := 2.$$

$$WS := 32$$

$$schmul := [[160,\ 0.00003],\ [3200,\ 0.007],\ [332224,\ 3.]]$$

$$classicmul := [[32,\ 2.500000000\ 10^{-7}],\ [1024,\ 0.0002560000000]]$$

$$classicsqr := [[32,\ 2.500000000\ 10^{-7}],\ [64,\ 7.875000000\ 10^{-7}],\ [128,\ 0.000002625000000],\ [256,\ 0.000008925000000],\ [512,\ 0.00003570000000],\ [1024,\ 0.0001386000000]]$$

$$karamul := [[128,\ 0.000004035000000],\ [256,\ 0.00001237500000],\ [512,\ 0.00004185000000],\ [1024,$$

0.0001356250000], [2048, 0.0004292500000], [4096,
0.001329750000], [8192, 0.004075000000], [16384,
0.01252000000], [32768, 0.03825000000], [65536,
0.1169500000], [131072, 0.3562500000], [262144,
1.077250000], [524288, 3.247000000], [1048576, 9.769750000]]

sqr := [[128, 0.000002655000000], [256, 0.000008710000000], [512,
0.00003110000000], [1024, 0.0001075500000], [2048,
0.0003580000000], [4096, 0.001156000000], [8192,
0.003662500000], [16384, 0.01147500000], [32768,
0.03592500000], [65536, 0.1106000000], [131072,
0.3397500000], [262144, 1.036000000], [524288,
3.148750000], [1048576, 9.540500000]]

karasqr := [[128, 0.000002722500000], [256,
0.000008612500000], [512, 0.00002945000000], [1024,
0.00009442500000], [2048, 0.0002985000000], [4096,
0.0009122500000], [8192, 0.002652500000], [16384,
0.008097500000], [32768, 0.02432250000], [65536,
0.07417500000], [131072, 0.2253000000], [262144,
0.6860390625], [524288, 2.067140625], [1048576, 6.203226562]]

fftmul := [[448, 0.0002490000000], [896, 0.0005295000000], [1792,
0.001136000000], [3584, 0.002447500000], [7168,
0.005300000000], [14336, 0.01171250000], [15360,
0.01288250000], [28672, 0.02605000000], [30720,
0.02790000000], [57344, 0.05780000000], [61440,
0.06037500000], [114688, 0.1251500000], [122880,
0.1288500000], [229376, 0.2740000000], [245760,
0.2735000000], [491520, 0.5787500000], [983040, 1.232750000]]

fftsqrold := [[448, 0.0001837750000], [896, 0.0003905000000], [1792,
0.0008302500000], [3584, 0.001774000000], [7168,
0.003862500000], [14336, 0.008265000000], [15360,
0.009670000000], [28672, 0.01814750000], [30720,
0.02052500000], [57344, 0.04105000000], [61440,

0.04512500000], [114688, 0.08947500000], [122880,
0.09590000000], [229376, 0.1991750000], [245760,
0.2037750000], [491520, 0.4355000000], [983040, 0.9357500000]]

fftsqr := [[448, 0.0001760750000], [896, 0.0003730000000], [1792,
0.0007927500000], [3584, 0.001697500000], [7168,
0.003650000000], [14336, 0.007830000000], [15360,
0.008585000000], [28672, 0.01725500000], [30720,
0.01833750000], [57344, 0.03910000000], [61440,
0.04060000000], [114688, 0.08475000000], [122880,
0.08677500000], [229376, 0.1856500000], [245760,
0.1841000000], [491520, 0.4007500000], [983040, 0.8382500000]]

rfffftmulc := [[704, 0.0003700000000], [1344, 0.0008007500000], [2688,
0.001601500000], [5120, 0.003585000000], [10240,
0.007552500000], [19456, 0.01759750000], [38912,
0.04202500000], [73728, 0.08862500000], [139264,
0.1851500000], [278528, 0.3897500000], [524288,
0.8080000000], [1048576, 1.686500000]]

rfffftmul2ss := [[704, 0.0001827750000], [1344,
0.0003587500000], [2688, 0.0007195000000], [5120,
0.001561250000], [10240, 0.003647500000], [19456,
0.008912500000], [38912, 0.02304000000], [73728,
0.04590000000], [139264, 0.1061500000], [278528,
0.2147250000], [524288, 0.4592500000], [1048576, 0.9460000000]]

rfffftsqrc := [[704, 0.0003207500000], [1344, 0.0006597500000], [2688,
0.001370250000], [5120, 0.002837500000], [10240,
0.006055000000], [19456, 0.01368750000], [38912,
0.03382500000], [73728, 0.07090000000], [139264,
0.1509750000], [278528, 0.3010000000], [524288,
0.6405000000], [1048576, 1.314500000]]

rfffftsqr2ss := [[704, 0.0001293750000], [1344,
0.0002610000000], [2688, 0.0004957500000], [5120,
0.001091500000], [10240, 0.002377750000], [19456,

(7.6.3)

```
0.006382500000], [38912, 0.01585500000], [73728,
0.03305000000], [139264, 0.07090000000], [278528,
0.1480750000], [524288, 0.3140000000], [1048576, 0.6462500000]]
```

```
> lTpb:=proc(T) local x;
  map(x->[log[b](x[1]),log[b](4*10^7*x[2]/x[1])],T);
  end;
  ppl:=proc(L)
  pointplot([log[b](L[1]),log[b](24*3600/L[2])],
  title=`Primality testing, log[2](test/day)-log[2](bit)`),
  textplot([log[b](L[1]),log[b](24*3600/L[2]),L[3]],align=
  {BELOW,LEFT})
  end;
  pplb:=proc(L)
  pointplot([log[b](L[1]),log[b](24*3600/L[2])],
  title=`Primality testing, log[2](test/day)-log[2](bit)`),
  textplot([log[b](L[1]),log[b](24*3600/L[2]),L[3]],align=
  {BELOW})
  end;
  pll:=proc(L) local x;
  plot(map(x->[log[b](x[1]),log[b](24*3600/x[2])],L))
  end;
  ltpb:=proc(L) local x;
  plot(map(x->[log[b](x[1]),log[b](24*3600/(f*x[2]*x[1]))],L));
  end;
```

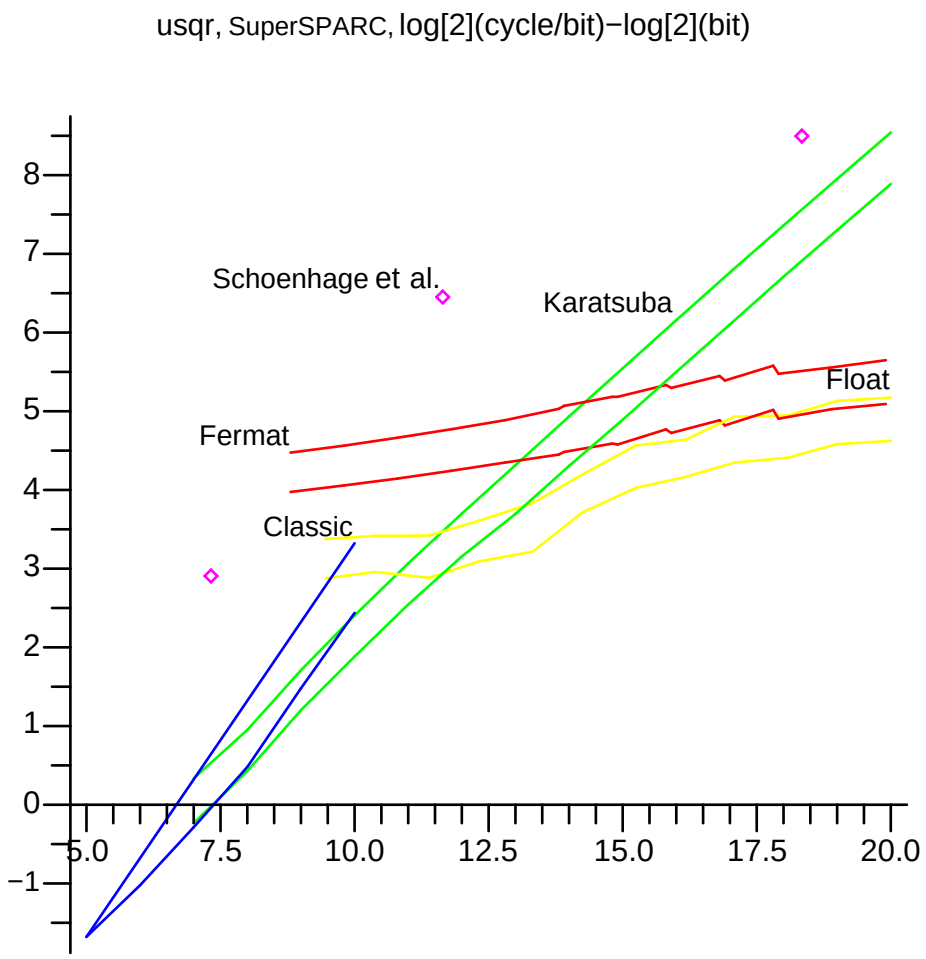
$$lTpb := \mathbf{proc}(T)$$
$$\quad \mathbf{local}\ x;$$
$$\quad map(\mathbf{proc}(x)$$
$$\qquad \mathbf{option}\ operator,\ arrow;$$
$$\qquad [\log[b](x[1]),\ \log[b](40000000 * x[2] / x[1])]$$
$$\quad \mathbf{end\ proc},\ T)$$
$$\mathbf{end\ proc}$$

$$ppl := \mathbf{proc}(L)$$
$$\quad pointplot([\log[b](L[1]),\ \log[b](86400 / L[2])],$$
$$\quad title = Primality\ testing,\ log[2](test/day) - log[2](bit)),$$
$$\quad textplot([\log[b](L[1]),\ \log[b](86400 / L[2]),\ L[3]],\ align = (\{BELOW,$$
$$\quad LEFT\}))$$
$$\mathbf{end\ proc}$$

$$pplb := \mathbf{proc}(L)$$
$$\quad pointplot([\log[b](L[1]),\ \log[b](86400 / L[2])],$$

```
    title = Primality testing, log[2](test/day)-log[2](bit)),
    textplot([log[b](L[1]), log[b](86400 / L[2]), L[3]], align = {BELOW})
 end proc
pll := proc(L)
    local x;
    plot(map(proc(x)
        option operator, arrow;
        [log[b](x[1]), log[b](86400 / x[2])]
    end proc, L))
 end proc
ltpb := proc(L)                                                    (7.6.4)
    local x;
    plot(map(proc(x)
        option operator, arrow;
        [log[b](x[1]), log[b](86400 / (f*x[2]*x[1]))]
    end proc, L))
 end proc
```

```
> display([plot(lTpb(schmul),style=POINT,title=`usqr,
  SuperSPARC, log[2](cycle/bit)-log[2](bit)`,color=magenta),
  textplot([lTpb(schmul)[2][1],lTpb(schmul)[2][2],
  `Schoenhage et al.`],align={ABOVE,LEFT}),
  plot(lTpb(karamul),color=green),
  plot(lTpb(karasqr),color=green),
  textplot([lTpb(karamul)[10][1],lTpb(karamul)[10][2],
  `Karatsuba`],align={ABOVE,LEFT}),
  plot(lTpb(classicmul),color=blue),
  plot(lTpb(classicsqr),color=blue),
  textplot([lTpb(classicmul)[2][1],lTpb(classicmul)[2][2],
  `Classic`],align={ABOVE,LEFT}),
  plot(lTpb(fftmul),color=red),
  plot(lTpb(fftsqr),color=red),
  textplot([lTpb(fftmul)[1][1],lTpb(fftmul)[1][2],`Fermat`],
  align={ABOVE,LEFT}),
  plot(lTpb(rffftmul2ss),color=yellow),
  plot(lTpb(rffftsqr2ss),color=yellow),
  textplot([lTpb(rffftmul2ss)[12][1],lTpb(rffftmul2ss)[12][2],
  `Float`],align={ABOVE,LEFT})]);
```




```
> f:=1.05*40/60;
  T1:=[log[2](663777)+7650,6,`Twin, Amdahl 1200, 1989`];
  T2:=[log[2](1706595)+11235,10,`Twin, Amdahl 1200, 1989`];
  T3:=[log[2](697053813)+16352,212,`Twin, SuperSPARC 60 MHz,
  1994`];
  T4:=[log[2](697053813)+16352,2*24*3600,`Maple, SuperSPARC 60
  MHz`];
  T5:=[log[2](697053813)+16352,42.5*60,`LiDIA, MIPS RS4000`];
  T6:=[log[2](697053813)+16352,3.5*3600,`LiDIA, SUN4 ELC`];
  T7:=[log[2](242206083)+38880,7*60,`Twin, SuperSPARC 60MHz,
  1995`];
  T8:=[log[2](242206083)+38880,5*3600,`LiDIA, SPARCstation20`];
  T:=[[WS*2^7,36.49/40000],[WS*2^8,10.61/4000],[19*2^10,25.53/
  (4*10^3)],
  [19*2^11,63.42/(4*10^3)],[18*2^12,13.22/400],[17*2^13,
  28.36/400],
  [17*2^14,59.23/400],[16*2^15,12.56/40],[16*2^16,25.85/40]];
  M29:=[110503,0.03*110503,`Prime, NEC SX-2, 1988`];
  M28:=[86243,5782,`Prime, CRAY-1, 1983`];
  M31:=[216091,3*3600,`Prime, 1 CPU CRAY XMP, 1985`];
```

```
M33:=[859433,25924,`Prime, 1 CPU CRAY C916, 1994`];
P1:=[216100,33*60,`Prime, Amdahl 1200 E, 1990`];
# P:=[[128000,128000*0.2],[256000,256000*0.5],[512000,512000]
];
display([ppl(T1),ppl(T2),ppl(T3),ppl(T4),ppl(T5),ppl(T6),ppl
(T7),ppl(T8),
ltpb(T),ppl(M28),ppl(M29),ppl(M31),pplb(M33),ppl(P1)]);
```

$$f := 0.7000000000$$

$$T1 := \left[\frac{\ln(663777)}{\ln(2)} + 7650,\ 6,\ \textit{Twin, Amdahl 1200, 1989}\right]$$

$$T2 := \left[\frac{\ln(1706595)}{\ln(2)} + 11235,\ 10,\ \textit{Twin, Amdahl 1200, 1989}\right]$$

$$T3 := \left[\frac{\ln(697053813)}{\ln(2)} + 16352,\ 212,\ \textit{Twin, SuperSPARC 60 MHz, 1994}\right]$$

$$T4 := \left[\frac{\ln(697053813)}{\ln(2)} + 16352,\ 172800,\ \textit{Maple, SuperSPARC 60 MHz}\right]$$

$$T5 := \left[\frac{\ln(697053813)}{\ln(2)} + 16352,\ 2550.0,\ \textit{LiDIA, MIPS RS4000}\right]$$

$$T6 := \left[\frac{\ln(697053813)}{\ln(2)} + 16352,\ 12600.0,\ \textit{LiDIA, SUN4 ELC}\right]$$

$$T7 := \left[\frac{\ln(242206083)}{\ln(2)} + 38880,\ 420,\ \textit{Twin, SuperSPARC 60MHz, 1995}\right]$$

$$T8 := \left[\frac{\ln(242206083)}{\ln(2)} + 38880,\ 18000,\ \textit{LiDIA, SPARCstation20}\right]$$

$$T := [[4096, 0.0009122500000], [8192, 0.002652500000], [19456, 0.006382500000], [38912, 0.01585500000], [73728, 0.03305000000], [139264, 0.07090000000], [278528, 0.1480750000], [524288, 0.3140000000], [1048576, 0.6462500000]]$$

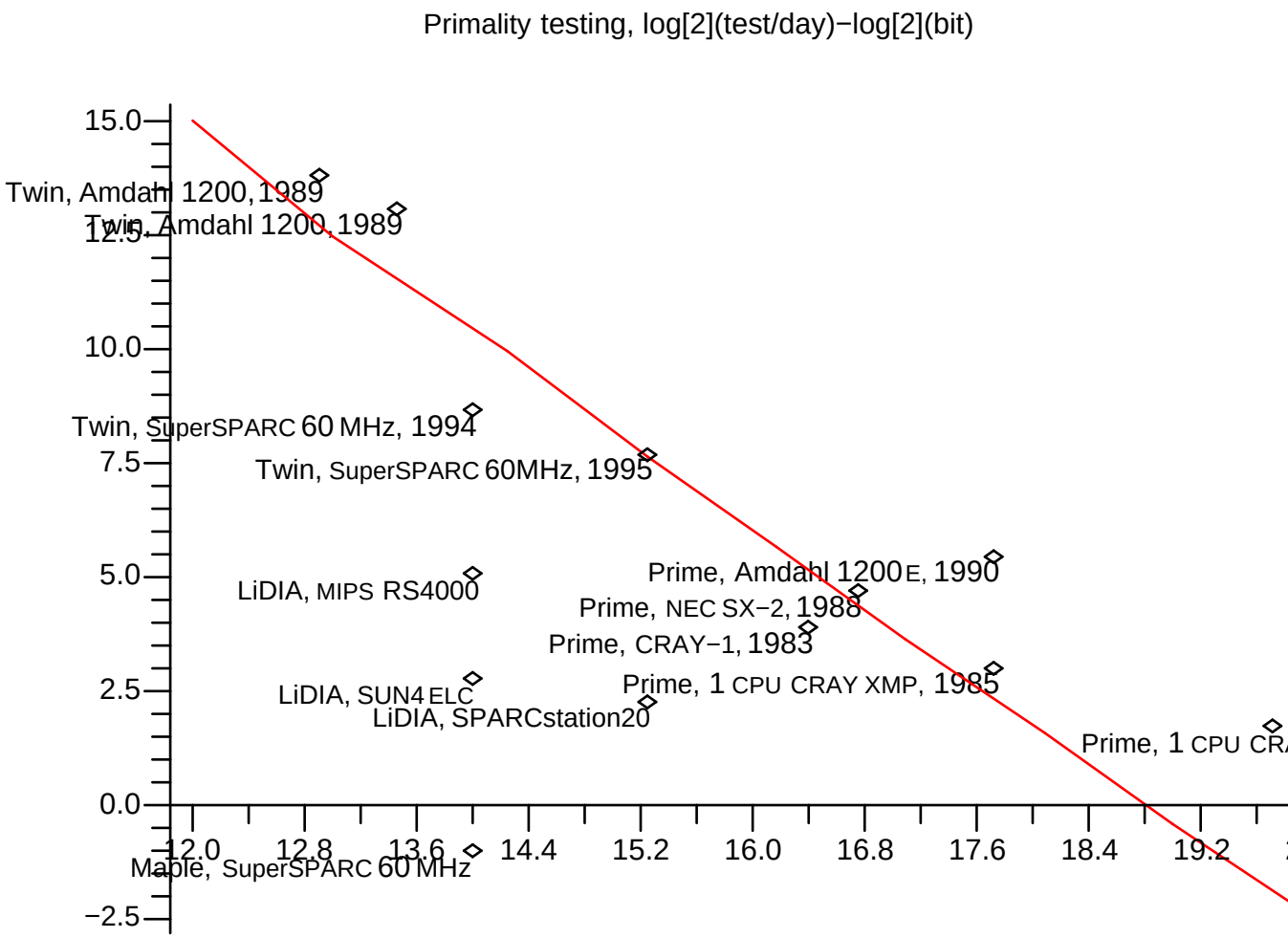
$$M29 := [110503,\ 3315.09,\ \textit{Prime, NEC SX-2, 1988}]$$

$$M28 := [86243,\ 5782,\ \textit{Prime, CRAY-1, 1983}]$$

$$M31 := [216091,\ 10800,\ \textit{Prime, 1 CPU CRAY XMP, 1985}]$$

$$M33 := [859433,\ 25924,\ \textit{Prime, 1 CPU CRAY C916, 1994}]$$

$$P1 := [216100,\ 1980,\ \textit{Prime, Amdahl 1200 E, 1990}]$$



7.7. Példa.

>

7.8. FFT véges testek felett.

>

7.9. Fermat-szám transzformáció.

>

7.10. Schönhage-Strassen-féle gyorsszorzó algoritmus.

>

7.11. Példa.

```
>
```

7.12. Ritka polinomok es ritka számok.

7.13. Feladat.

7.14. Osztás, polinomosztás.

```
> #
  # This procedure calculate the approximate reciprocal b of
  # a given number a. The binary length of a must be 2^logn+1,
  # where logn>2 is an integer. The result is [b,alpha,beta]
  # where the parameters 0<alpha<1/2 and beta<=-2 are such that
  # with the notation n=2^logn we have a*b=2^(2*n)-s with
  # 0<s<=a*2^(alpha*n+beta).
  #

  apprec:=proc(a::posint,logn::posint) local n,b,ai,bi,L;
  n:=2^logn;
  if logn<=5 then
    b:=floor(2^(2*n)/a);
    if a*b=2^(2*n) then b:=b-1 fi;
    RETURN([b,2/n,-2]);
  fi;
  ai:=iquo(a,2^(n/2));
  L:=apprec(ai,logn-1);
  bi:=L[1];
  iquo(2^(3*n/2+1)*bi-bi*bi*a,2^n);
  [%,L[2],L[3]]
  end;
```

$$apprec := \mathbf{proc}(a::posint,\ logn::posint) \tag{7.14.1}$$

```
local n, b, ai, bi, L;
n := 2 ^ logn;
if logn <= 5 then
    b := floor(2 ^ (2 * n) / a);
    if a * b = 2 ^ (2 * n) then
        b := b - 1
    end if;
    RETURN([b, 2 / n, -2])
end if;
```

```
    ai:= iquo(a, 2^(1/2*n));
    L:= apprec(ai, logn-1);
    bi:= L[1];
    iquo(2^(3/2*n+1)*bi-bi*bi*a, 2^n);
    [%, L[2], L[3]]
end proc
```

```
> apprec(256,3); apprec(257,3); apprec(300,3);
  apprec(511,3); apprec(65536,4); apprec(65537,4);
```

$$\left[255, \frac{1}{4}, -2\right]$$

$$\left[255, \frac{1}{4}, -2\right]$$

$$\left[218, \frac{1}{4}, -2\right]$$

$$\left[128, \frac{1}{4}, -2\right]$$

$$\left[65535, \frac{1}{8}, -2\right]$$

$$\left[65535, \frac{1}{8}, -2\right] \qquad (7.14.2)$$

```
> #
  # This is a Maple demo program to show the procedure
  # for division using the approximation of reciprocal
  # based on Newton's method. Here, with the notation
  # n=2^logn, c has at most 2n digit and a has at most
  # n digit. logn must be an integer greater then 1.
  #

  division:=proc(c::posint,a::posint,logn::posint)
  local n,k,l,as,bs,cs,qs,rs,alpha;
  n:=2^logn; as:=a;
  for k from 0 while as<2^n do as:=2*as od;
  if c=0 then RETURN([0,0]) fi; cs:=floor(c/2);
  for l from -1 while cs<2^(2*n-1) do cs:=2*cs od;
  print(l); print(k);
  if k<=0 or l<0 or k>l+1 then RETURN(`overflow`) fi;
  cs:=iquo(cs,2^n); apprec(as,logn);
  bs:=%[1]; alpha:=%%[2]; qs:=iquo(cs*bs,2^(n-k+l)); rs:=c-qs*a;
  if k-l+alpha*n>1 then division(rs,a,logn); qs:=qs+%[1]; rs:=
  %%[2]; fi;
  while rs>=a do qs:=qs+1; rs:=rs-a; od;
```

```
    if qs>=2^n then RETURN(`overflow`) fi;
    [qs,rs];
    end;
```

$$division := \mathbf{proc}(c::posint,\ a::posint,\ logn::posint) \tag{7.14.3}$$

```
    local n, k, l, as, bs, cs, qs, rs, α;
    n := 2^logn;
    as := a;
    for k from 0 while as < 2^n do
        as := 2 * as
    end do;
    if c = 0 then
        RETURN([0, 0])
    end if;
    cs := floor(1 / 2 * c);
    for l from -1 while cs < 2^(2 * n - 1) do
        cs := 2 * cs
    end do;
    print(l);
    print(k);
    if k <= 0 or l < 0 or l + 1 < k then
        RETURN(overflow)
    end if;
    cs := iquo(cs, 2^n);
    apprec(as, logn);
    bs := %[1];
    α := %%[2];
    qs := iquo(cs * bs, 2^(n - k + l));
    rs := c - qs * a;
    if 1 < k - l + α * n then
        division(rs, a, logn);
        qs := qs + %[1];
        rs := %%[2]
    end if;
```

```
    while a <= rs do
        qs := qs + 1;
        rs := rs - a
    end do;
    if 2^n <= qs then
        RETURN(overflow)
    end if;
    [qs, rs]
end proc
```

```
> division(2^32,2^16,4);
```

$$-1$$

$$0$$

$$overflow \tag{7.14.4}$$

```
> debug(division);
```

$$division \tag{7.14.5}$$

```
> division(2^32-10*2^16,2^16-1,4);
{--> enter division, args = 4294311936, 65535, 4
```

$$n := 16$$

$$as := 65535$$

$$as := 131070$$

$$cs := 2147155968$$

$$cs := 4294311936$$

$$0$$

$$1$$

$$cs := 65526$$

$$\left[32768, \frac{1}{8}, -2\right]$$

$$bs := 32768$$

$$\alpha := \frac{1}{8}$$

$$qs := 65526$$

$$rs := 65526$$

```
{--> enter division, args = 65526, 65535, 4
```

$$n := 16$$

$$as := 65535$$

$$as := 131070$$

$$cs := 32763$$

$$cs := 65526$$

$$cs := 131052$$

$$cs := 262104$$

$$cs := 524208$$

$$cs := 1048416$$

$$cs := 2096832$$

$$cs := 4193664$$

$$cs := 8387328$$

$$cs := 16774656$$

$$cs := 33549312$$

$$cs := 67098624$$

$$cs := 134197248$$

$$cs := 268394496$$

$$cs := 536788992$$

$$cs := 1073577984$$

$$cs := 2147155968$$

$$cs := 4294311936$$

$$16$$

$$1$$

$$cs := 65526$$

$$\left[32768,\ \frac{1}{8},\ -2\right]$$

$$bs := 32768$$

$$\alpha := \frac{1}{8}$$

$$qs := 0$$

$$rs := 65526$$

$$[0,\ 65526]$$

```
<-- exit division (now in division) = [0, 65526]}
```

$$[0,\ 65526]$$

$$qs := 65526$$

$$rs := 65526$$

$$[65526, 65526]$$

```
<-- exit division (now at top level) = [65526, 65526]}
```

$$[65526, 65526] \tag{7.14.6}$$

```
> %[1]*(2^16-1)+%[2]-(2^32-10*2^16);
```

$$0 \tag{7.14.7}$$

7.15. Polinom kiértékelése tetszőleges helyeken.

```
>
```

7.16. Interpoláció.

7.17. Feladat.

7.18. Feladat.

7.19. Feladat.

7.20. Feladat: kontrollált euklidészi leszállás.

```
> # This function makes an upper or lower division step. The
  # function updapes the incoming list [a,b,x,y,u,v], but the
  # remainder is given back.

  qstep:=proc(L,s) local q,r,LL;
    if L[1]>L[2] then
      q:=iquo(L[1],L[2]); r:=irem(L[1],L[2]);
      if r>=s then
        LL:=[r,L[2],L[3],L[4]+q*L[3],L[5],L[6]+q*L[5]];
      else
        LL:=[r,L[2],L[3],L[4]+(q-1)*L[3],L[5],L[6]+(q-1)*L[5]];
      fi;
    else
      q:=iquo(L[2],L[1]); r:=irem(L[2],L[1]);
      if r>=s then
        LL:=[L[1],r,L[3]+q*L[4],L[4],L[5]+q*L[6],L[6]];
      else
        LL:=[L[1],r,L[3]+(q-1)*L[4],L[4],L[5]+(q-1)*L[6],L[6]];
      fi;
    fi;
    LL;
```

```
    end;
```

```
qstep := proc(L, s)                                                    (7.20.1)
    local q, r, LL;
    if L[2] < L[1] then
        q := iquo(L[1], L[2]);
        r := irem(L[1], L[2]);
        if s <= r then
            LL := [r, L[2], L[3], L[4] + q*L[3], L[5], L[6] + q*L[5]]
        else
            LL := [r, L[2], L[3], L[4] + (q - 1)*L[3], L[5], L[6]
            + (q - 1)*L[5]]
        end if
    else
        q := iquo(L[2], L[1]);
        r := irem(L[2], L[1]);
        if s <= r then
            LL := [L[1], r, L[3] + q*L[4], L[4], L[5] + q*L[6], L[6]]
        else
            LL := [L[1], r, L[3] + (q - 1)*L[4], L[4], L[5]
            + (q - 1)*L[6], L[6]]
        end if
    end if;
    LL
end proc
```

```
> # This is a slow version of ced.  It works for arbitrary
  input.
  # The matrix and the remainder is given back.

  sced:=proc(a,b,s) local L;
    L:=[a,b,1,0,0,1];
    while L[1]>=s and L[2]>=s do
      L:=qstep(L,s);
    od;
    L;
  end;
```

```
sced := proc(a, b, s)                                                  (7.20.2)
```

```
    local L;
    L := [a, b, 1, 0, 0, 1];
    while s <= L[1] and s <= L[2] do
        L := qstep(L, s)
    end do;
    L
end proc
```

```
> debug(sced);
```

$$sced \tag{7.20.3}$$

```
> a:=floor(evalf(Pi*10.^8)); b:=floor(evalf(exp(1.)*10^8));
  sced(a,b,1);
```

$$a := 314159265$$

$$b := 271828182$$

```
{--> enter sced, args = 314159265, 271828182, 1
```

$$L := [314159265, 271828182, 1, 0, 0, 1]$$

$$L := [42331083, 271828182, 1, 1, 0, 1]$$

$$L := [42331083, 17841684, 7, 1, 6, 1]$$

$$L := [6647715, 17841684, 7, 15, 6, 13]$$

$$L := [6647715, 4546254, 37, 15, 32, 13]$$

$$L := [2101461, 4546254, 37, 52, 32, 45]$$

$$L := [2101461, 343332, 141, 52, 122, 45]$$

$$L := [41469, 343332, 141, 898, 122, 777]$$

$$L := [41469, 11580, 7325, 898, 6338, 777]$$

$$L := [6729, 11580, 7325, 22873, 6338, 19791]$$

$$L := [6729, 4851, 30198, 22873, 26129, 19791]$$

$$L := [1878, 4851, 30198, 53071, 26129, 45920]$$

$$L := [1878, 1095, 136340, 53071, 117969, 45920]$$

$$L := [783, 1095, 136340, 189411, 117969, 163889]$$

$$L := [783, 312, 325751, 189411, 281858, 163889]$$

$$L := [159, 312, 325751, 840913, 281858, 727605]$$

$$L := [159, 153, 1166664, 840913, 1009463, 727605]$$

$$L := [6, 153, 1166664, 2007577, 1009463, 1737068]$$

$$L := [6, 3, 51356089, 2007577, 44436163, 1737068]$$

$$L := [0, 3, 51356089, 53363666, 44436163, 46173231]$$

$$[0, 3, 51356089, 53363666, 44436163, 46173231]$$

```
<-- exit sced (now at top level) = [0, 3, 51356089,
53363666, 44436163, 46173231]}
```

$$[0, 3, 51356089, 53363666, 44436163, 46173231] \tag{7.20.4}$$

```
> 3*(%[3]+%[4]),3*(%[5]+%[6]),%[3]*%[6]-%[4]*%[5];
```

$$314159265, 271828182, 1 \tag{7.20.5}$$

```
> lB:=4; B:=2^lB; # wordsize and base of number system
```

$$lB := 4$$

$$B := 16 \tag{7.20.6}$$

```
> cedmulQM:=proc(Q,M) local i,x,y,u,v,q;
  x:=M[1]; y:=M[2]; u:=M[3]; v:=M[4];
  for i from nops(Q) by -1 while i>0 do
    q:=Q[i];
    if q[1]=0 then
      q:=q[2]; u:=u+q*x; v:=v+q*y;
    else
      q:=q[1]; x:=x+q*u; y:=y+q*v;
    fi;
  od;
  [x,y,u,v];
  end;
```

$$cedmulQM := \mathbf{proc}(Q, M) \tag{7.20.7}$$

$$\begin{aligned}
&\mathbf{local}\ i, x, y, u, v, q;\\
&x := M[1];\\
&y := M[2];\\
&u := M[3];\\
&v := M[4];\\
&\mathbf{for}\ i\ \mathbf{from}\ nops(Q)\ \mathbf{by}\ -1\ \mathbf{while}\ 0 < i\ \mathbf{do}\\
&\quad q := Q[i];\\
&\quad \mathbf{if}\ q[1] = 0\ \mathbf{then}\\
&\quad\quad q := q[2];\\
&\quad\quad u := u + q * x;\\
&\quad\quad v := v + q * y\\
&\quad \mathbf{else}\\
&\quad\quad q := q[1];\\
&\quad\quad x := x + q * u;\\
&\quad\quad y := y + q * v
\end{aligned}$$

```
        end if
    end do;
    [x, y, u, v]
end proc
```

```
> cedmulMQ:=proc(M,Q) local i,x,y,u,v,q;
  x:=M[1]; y:=M[2]; u:=M[3]; v:=M[4];
  for i while i<=nops(Q) do
    q:=Q[i];
    if q[1]=0 then
      q:=q[2]; x:=x+q*y; u:=u+q*v;
    else
      q:=q[1]; y:=y+q*x; v:=v+q*u;
    fi;
  od;
  [x,y,u,v];
  end;
```

$$cedmulMQ := \mathbf{proc}(M, Q) \tag{7.20.8}$$

```
    local i, x, y, u, v, q;
    x := M[1];
    y := M[2];
    u := M[3];
    v := M[4];
    for i while i <= nops(Q) do
        q := Q[i];
        if q[1] = 0 then
            q := q[2];
            x := x + q*y;
            u := u + q*v
        else
            q := q[1];
            y := y + q*x;
            v := v + q*u
        end if
    end do;
    [x, y, u, v]
end proc
```

```
> cedmulMM:=proc(M,MM) local x,y,u,v;
  x:=M[1]*MM[1]+M[2]*MM[3];
  y:=M[1]*MM[2]+M[2]*MM[4];
  u:=M[3]*MM[1]+M[4]*MM[3];
  v:=M[3]*MM[2]+M[4]*MM[4];
  [x,y,u,v];
  end;
```

$$cedmulMM := \mathbf{proc}(M, MM) \tag{7.20.9}$$

$\mathbf{local}\, x, y, u, v;$

$x := M[1]*MM[1] + M[2]*MM[3];$

$y := M[1]*MM[2] + M[2]*MM[4];$

$u := M[3]*MM[1] + M[4]*MM[3];$

$v := M[3]*MM[2] + M[4]*MM[4];$

$[x, y, u, v]$

end proc

```
> cedret:=proc(M,L) local x,y,u,v,q,MM,LL;
  x:=M[1]; y:=M[2]; u:=M[3]; v:=M[4];
  if L[4]=0 then
    q:=L[5]; x:=x+q*y; u:=u+q*v;
  else
    q:=L[4];
    y:=y+q*x; v:=v+q*u;
  fi;
  [L[1],L[2],x,y,u,v];
  end;
```

$$cedret := \mathbf{proc}(M, L) \tag{7.20.10}$$

$\mathbf{local}\, x, y, u, v, q, MM, LL;$

$x := M[1];$

$y := M[2];$

$u := M[3];$

$v := M[4];$

$\mathbf{if}\, L[4] = 0\ \mathbf{then}$

$q := L[5];$

$x := x + q*y;$

$u := u + q*v$

else

$q := L[4];$

$y := y + q*x;$

```
        v:= v + q*u
    end if;
    [L[1], L[2], x, y, u, v]
end proc
```

```
> # Set the minimal positive integer l such that u,v<B^l

  min_l:=proc(u,v) local l,l0,l1;
    l0:=1; l1:=1;
    while u>=B^l1 or v>=B^l1 do l0:=l1; l1:=2*l1; od;
    l:=ceil((l0+l1)/2);
    while l<l1 do
      if u<B^l and v<B^l then l1:=l else l0:=l fi;
      l:=ceil((l0+l1)/2);
    od; l;
  end;
```

$$min\_l := \mathbf{proc}(u, v) \tag{7.20.11}$$

```
    local l, l0, l1;
    l0 := 1;
    l1 := 1;
    while B^l1 <= u or B^l1 <= v do
        l0 := l1;
        l1 := 2 * l1
    end do;
    l := ceil(1 / 2 * l0 + 1 / 2 * l1);
    while l < l1 do
        if u < B^l and v < B^l then
            l1 := l
        else
            l0 := l
        end if;
        l := ceil(1 / 2 * l0 + 1 / 2 * l1)
    end do;
    l
end proc
```

```
> min_l(a,b);
```


(7.20.12)

```
> ced:=proc(a,b,n,w,e) local nn,np,ep,wp,aa,bb,ap,bp,app,bpp,f,
  s,L,M,MM,Q,t;
  s:=w*B^e;
  if n<=2 then return(sced(a,b,s)) fi;
  nn:=n; aa:=a; bb:=b;
  ep:=ceil(nn/4); np:=2*ep;
  if nn-ep<e then ep:=nn-e; np:=2*ep fi;
  f:=B^(nn-np);
  ap:=floor(aa/f); bp:=floor(bb/f); app:=aa-ap*f; bpp:=bb-bp*f;
  if e+ep=nn then wp:=w+1 else wp:=1;
    if e>=ep and e>=nn-3*ep then
      if ap*f+w*B^(e+ep)>B^nn or bp*f+w*B^(e+ep)>B^nn then wp:=
  2; fi;
    fi;
  fi;
  if ap>=wp*B^ep and bp>=wp*B^ep then
    L:=ced(ap,bp,np,wp,ep);
    M:=[L[3],L[4],L[5],L[6]];
    aa:=L[1]; bb:=L[2]; if aa<bb then aa:=aa+bb else bb:=bb+aa
  fi;
    aa:=aa*f+M[4]*app-M[2]*bpp; bb:=bb*f-M[3]*app+M[1]*bpp;
  else
    M:=[1,0,0,1];
  fi;
  t:=M[1]*aa+M[2]*bb; t:=M[3]*aa+M[4]*bb; t:=M[1]*M[4]-M[2]*M
  [3]; # test
  Q:=[];
  while aa>=f*B^ep or bb>=f*B^ep do
    L:=qstep([aa,bb,1,0,0,1],s);
    aa:=L[1]; bb:=L[2];
    if aa<s or bb<s then
      L:=cedret(cedmulMQ(M,Q),L);
      aa:=L[1]; bb:=L[2]; if aa<bb then aa:=aa+bb else bb:=bb+
  aa fi; # test
      t:=L[3]*aa+L[4]*bb;t:=L[5]*aa+L[6]*bb;t:=L[3]*L[6]-L[4]*L
  [5]; # test
      return(L);
    fi;
    Q:=[op(Q),[L[4],L[5]]];
  od;
  MM:=cedmulMQ(M,Q); #test
  t:=MM[1]*aa+MM[2]*bb;t:=MM[3]*aa+MM[4]*bb;t:=MM[1]*MM[4]-MM
  [2]*MM[3]; # test
  nn:=nn-ep;
  if nn-ep<e then ep:=nn-e; np:=2*ep fi;
  f:=B^(nn-np);
  ap:=floor(aa/f); bp:=floor(bb/f); app:=aa-ap*f; bpp:=bb-bp*f;
  if e+ep=nn then wp:=w+1 else wp:=1;
```

```
    if e>=ep and e>=nn-3*ep then
      if ap*f+w*B^(e+ep)>B^nn or bp*f+w*B^(e+ep)>B^nn then wp:=
  2; fi;
    fi;
  fi;
  if ap>=wp*B^ep and bp>=wp*B^ep then
    L:=ced(ap,bp,np,wp,ep);
    MM:=[L[3],L[4],L[5],L[6]];
    aa:=L[1]; bb:=L[2]; if aa<bb then aa:=aa+bb else bb:=bb+aa
  fi;
    aa:=aa*f+MM[4]*app-MM[2]*bpp; bb:=bb*f-MM[3]*app+MM[1]*bpp;
  else
    MM:=[1,0,0,1];
  fi;
  MM:=cedmulQM(Q,MM); M:=cedmulMM(M,MM);
  t:=M[1]*aa+M[2]*bb; t:=M[3]*aa+M[4]*bb; t:=M[1]*M[4]-M[2]*M
  [3]; # test
  Q:=[];
  while aa>=f*B^ep or bb>=f*B^ep do
    L:=qstep([aa,bb,1,0,0,1],s);
    aa:=L[1]; bb:=L[2];
    if aa<s or bb<s then
      L:=cedret(cedmulMQ(M,Q),L);
      aa:=L[1]; bb:=L[2]; if aa<bb then aa:=aa+bb else bb:=bb+
  aa fi; # test
      t:=L[3]*aa+L[4]*bb;t:=L[5]*aa+L[6]*bb;t:=L[3]*L[6]-L[4]*L
  [5]; # test
      return(L);
    fi;
    Q:=[op(Q),[L[4],L[5]]];
  od;
  MM:=cedmulMQ(M,Q); #test
  t:=MM[1]*aa+MM[2]*bb;t:=MM[3]*aa+MM[4]*bb;t:=MM[1]*MM[4]-MM
  [2]*MM[3]; # test
  L:=ced(aa,bb,nn-ep,w,e);
  MM:=[L[3],L[4],L[5],L[6]];
  MM:=cedmulQM(Q,MM); M:=cedmulMM(M,MM);
  aa:=L[1]; bb:=L[2]; if aa<bb then aa:=aa+bb else bb:=bb+aa
  fi; # test
  t:=M[1]*aa+M[2]*bb;t:=M[3]*aa+M[4]*bb;t:=M[1]*M[4]-M[2]*M[3];
  # test
  [L[1],L[2],M[1],M[2],M[3],M[4]];
  end;
```

$ced := \mathbf{proc}(a, b, n, w, e)$ (7.20.13)

$\mathbf{local}\ nn, np, ep, wp, aa, bb, ap, bp, app, bpp, f, s, L, M, MM, Q, t;$

$s := w * B\hat{}e;$

$\mathbf{if}\ n <= 2\ \mathbf{then}$

```
        return sced(a, b, s)
    end if;
    nn := n;
    aa := a;
    bb := b;
    ep := ceil(1 / 4 * nn);
    np := 2 * ep;
    if nn − ep < e then
        ep := nn − e;
        np := 2 * ep
    end if;
    f := B^(nn − np);
    ap := floor(aa / f);
    bp := floor(bb / f);
    app := aa − ap * f;
    bpp := bb − bp * f;
    if e + ep = nn then
        wp := w + 1
    else
        wp := 1;
        if ep <= e and nn − 3 * ep <= e then
            if B^nn < ap * f + w * B^(e + ep) or B^nn < bp * f + w * B^(e
            + ep) then
                wp := 2
            end if
        end if
    end if;
    if wp * B^ep <= ap and wp * B^ep <= bp then
        L := ced(ap, bp, np, wp, ep);
        M := [L[3], L[4], L[5], L[6]];
        aa := L[1];
        bb := L[2];
```

```
        if aa < bb then
            aa := aa + bb
        else
            bb := aa + bb
        end if;
        aa := aa* f + M[4]* app − M[2]* bpp;
        bb := bb* f − M[3]* app + M[1]* bpp
    else
        M := [1, 0, 0, 1]
    end if;
    t := M[1]* aa + M[2]* bb;
    t := M[3]* aa + M[4]* bb;
    t := M[1]* M[4] − M[2]* M[3];
    Q := [ ];
    while f* B^ep <= aa or f* B^ep <= bb do
        L := qstep([aa, bb, 1, 0, 0, 1], s);
        aa := L[1];
        bb := L[2];
        if aa < s or bb < s then
            L := cedret(cedmulMQ(M, Q), L);
            aa := L[1];
            bb := L[2];
            if aa < bb then
                aa := aa + bb
            else
                bb := aa + bb
            end if;
            t := L[3]* aa + L[4]* bb;
            t := L[5]* aa + L[6]* bb;
            t := L[3]* L[6] − L[4]* L[5];
            return L
        end if;
```

```
        Q := [op(Q), [L[4], L[5]]]
    end do;
    MM := cedmulMQ(M, Q);
    t := MM[1]*aa + MM[2]*bb;
    t := MM[3]*aa + MM[4]*bb;
    t := MM[1]*MM[4] - MM[2]*MM[3];
    nn := nn - ep;
    if nn - ep < e then
        ep := nn - e;
        np := 2*ep
    end if;
    f := B^(nn - np);
    ap := floor(aa / f);
    bp := floor(bb / f);
    app := aa - ap*f;
    bpp := bb - bp*f;
    if e + ep = nn then
        wp := w + 1
    else
        wp := 1;
        if ep <= e and nn - 3*ep <= e then
            if B^nn < ap*f + w*B^(e + ep) or B^nn < bp*f + w*B^(e
            + ep) then
                wp := 2
            end if
        end if
    end if;
    if wp*B^ep <= ap and wp*B^ep <= bp then
        L := ced(ap, bp, np, wp, ep);
        MM := [L[3], L[4], L[5], L[6]];
        aa := L[1];
        bb := L[2];
```

```
        if aa < bb then
            aa := aa + bb
        else
            bb := aa + bb
        end if;
        aa := aa* f + MM[4]* app − MM[2]* bpp;
        bb := bb* f − MM[3]* app + MM[1]* bpp
    else
        MM := [1, 0, 0, 1]
    end if;
    MM := cedmulQM(Q, MM);
    M := cedmulMM(M, MM);
    t := M[1]* aa + M[2]* bb;
    t := M[3]* aa + M[4]* bb;
    t := M[1]* M[4] − M[2]* M[3];
    Q := [ ];
    while f* B^ep <= aa or f* B^ep <= bb do
        L := qstep([aa, bb, 1, 0, 0, 1], s);
        aa := L[1];
        bb := L[2];
        if aa < s or bb < s then
            L := cedret(cedmulMQ(M, Q), L);
            aa := L[1];
            bb := L[2];
            if aa < bb then
                aa := aa + bb
            else
                bb := aa + bb
            end if;
            t := L[3]* aa + L[4]* bb;
            t := L[5]* aa + L[6]* bb;
            t := L[3]* L[6] − L[4]* L[5];
```

```
            return L
        end if;
        Q := [op(Q), [L[4], L[5]]]
    end do;
    MM := cedmulMQ(M, Q);
    t := MM[1]*aa + MM[2]*bb;
    t := MM[3]*aa + MM[4]*bb;
    t := MM[1]*MM[4] - MM[2]*MM[3];
    L := ced(aa, bb, nn - ep, w, e);
    MM := [L[3], L[4], L[5], L[6]];
    MM := cedmulQM(Q, MM);
    M := cedmulMM(M, MM);
    aa := L[1];
    bb := L[2];
    if aa < bb then
        aa := aa + bb
    else
        bb := aa + bb
    end if;
    t := M[1]*aa + M[2]*bb;
    t := M[3]*aa + M[4]*bb;
    t := M[1]*M[4] - M[2]*M[3];
    [L[1], L[2], M[1], M[2], M[3], M[4]]
end proc
```

```
> debug(ced);
```

$$ced \tag{7.20.14}$$

```
> ced(a,b,min_l(a,b),1,0);
{--> enter ced, args = 314159265, 271828182, 8, 1, 0
```

$$s := 1$$

$$nn := 8$$

$$aa := 314159265$$

$$bb := 271828182$$

$$ep := 2$$

$$np := 4$$

$$f := 65536$$

$$ap := 4793$$

$$bp := 4147$$

$$app := 45217$$

$$bpp := 50390$$

$$wp := 1$$

```
{--> enter ced, args = 4793, 4147, 4, 1, 2
```

$$s := 256$$

$$nn := 4$$

$$aa := 4793$$

$$bb := 4147$$

$$ep := 1$$

$$np := 2$$

$$f := 256$$

$$ap := 18$$

$$bp := 16$$

$$app := 185$$

$$bpp := 51$$

$$wp := 1$$

```
{--> enter ced, args = 18, 16, 2, 1, 1
```

$$s := 16$$

```
{--> enter sced, args = 18, 16, 16
```

$$L := [18, 16, 1, 0, 0, 1]$$

$$L := [2, 16, 1, 0, 0, 1]$$

$$[2, 16, 1, 0, 0, 1]$$

```
<-- exit sced (now in ced) = [2, 16, 1, 0, 0, 1]}
<-- exit ced (now in ced) = [2, 16, 1, 0, 0, 1]}
```

$$L := [2, 16, 1, 0, 0, 1]$$

$$M := [1, 0, 0, 1]$$

$$aa := 2$$

$$bb := 16$$

$$aa := 18$$

$$aa := 4793$$

$$bb := 4147$$

$$t := 4793$$

$$t := 4147$$

$$t := 1$$

$$Q := [\,]$$

$$L := [646, 4147, 1, 1, 0, 1]$$

$$aa := 646$$

$$bb := 4147$$

$$Q := [[1, 0]]$$

$$L := [646, 271, 1, 0, 6, 1]$$

$$aa := 646$$

$$bb := 271$$

$$Q := [[1, 0], [0, 6]]$$

$$MM := [7, 1, 6, 1]$$

$$t := 4793$$

$$t := 4147$$

$$t := 1$$

$$nn := 3$$

$$f := 16$$

$$ap := 40$$

$$bp := 16$$

$$app := 6$$

$$bpp := 15$$

$$wp := 2$$

$$MM := [1, 0, 0, 1]$$

$$MM := [7, 1, 6, 1]$$

$$M := [7, 1, 6, 1]$$

$$t := 4793$$

$$t := 4147$$

$$t := 1$$

$$Q := [\,]$$

$$L := [104, 271, 1, 1, 0, 1]$$

$$aa := 104$$

$$bb := 271$$

$$L := [104, 271, 7, 8, 6, 7]$$

$$aa := 104$$

$$bb := 271$$

$$aa := 375$$

$$t := 4793$$

$$t := 4147$$

$$t := 1$$

```
<-- exit ced (now in ced) = [104, 271, 7, 8, 6, 7]}
```

$$L := [104, 271, 7, 8, 6, 7]$$

$$M := [7, 8, 6, 7]$$

$$aa := 104$$

$$bb := 271$$

$$aa := 375$$

$$aa := 24489399$$

$$bb := 17841684$$

$$t := 314159265$$

$$t := 271828182$$

$$t := 1$$

$$Q := [\,]$$

$$L := [6647715, 17841684, 1, 1, 0, 1]$$

$$aa := 6647715$$

$$bb := 17841684$$

$$Q := [[1, 0]]$$

$$L := [6647715, 4546254, 1, 0, 2, 1]$$

$$aa := 6647715$$

$$bb := 4546254$$

$$Q := [[1, 0], [0, 2]]$$

$$MM := [37, 15, 32, 13]$$

$$t := 314159265$$

$$t := 271828182$$

$$t := 1$$

$$nn := 6$$

$$f := 256$$

$$ap := 25967$$

$$bp := 17758$$

$$app := 163$$

$$bpp := 206$$

$$wp := 1$$

```
{--> enter ced, args = 25967, 17758, 4, 1, 2
```

$$s := 256$$

$$nn := 4$$

$$aa := 25967$$

$$bb := 17758$$

$$ep := 1$$

$$np := 2$$

$$f := 256$$

$$ap := 101$$

$$bp := 69$$

$$app := 111$$

$$bpp := 94$$

$$wp := 1$$

```
{--> enter ced, args = 101, 69, 2, 1, 1
```

$$s := 16$$

```
{--> enter sced, args = 101, 69, 16
```

$$L := [101, 69, 1, 0, 0, 1]$$

$$L := [32, 69, 1, 1, 0, 1]$$

$$L := [32, 5, 2, 1, 1, 1]$$

$$[32, 5, 2, 1, 1, 1]$$

```
<-- exit sced (now in ced) = [32, 5, 2, 1, 1, 1]}
<-- exit ced (now in ced) = [32, 5, 2, 1, 1, 1]}
```

$$L := [32, 5, 2, 1, 1, 1]$$

$$M := [2, 1, 1, 1]$$

$$aa := 32$$

$$bb := 5$$

$$bb := 37$$

$$aa := 8209$$

$$bb := 9549$$

$$t := 25967$$

$$t := 17758$$

$$t := 1$$

$$Q := [\,]$$

$$L := [8209, 1340, 1, 0, 1, 1]$$

$$aa := 8209$$

$$bb := 1340$$

$$Q := [[0, 1]]$$

$$L := [169, 1340, 1, 5, 0, 1]$$

$$aa := 169$$

$$bb := 1340$$

$$L := [169, 1340, 3, 16, 2, 11]$$

$$aa := 169$$

$$bb := 1340$$

$$aa := 1509$$

$$t := 25967$$

$$t := 17758$$

$$t := 1$$

```
<-- exit ced (now in ced) = [169, 1340, 3, 16, 2, 11]}
```

$$L := [169, 1340, 3, 16, 2, 11]$$

$$MM := [3, 16, 2, 11]$$

$$aa := 169$$

$$bb := 1340$$

$$aa := 1509$$

$$aa := 384801$$

$$bb := 343332$$

$$MM := [11, 59, 8, 43]$$

$$M := [141, 757, 122, 655]$$

$$t := 314159265$$

$$t := 271828182$$

$$t := 1$$

$$Q := [\,]$$
$$L := [41469, 343332, 1, 1, 0, 1]$$
$$aa := 41469$$
$$bb := 343332$$
$$Q := [[1, 0]]$$
$$L := [41469, 11580, 1, 0, 8, 1]$$
$$aa := 41469$$
$$bb := 11580$$
$$Q := [[1, 0], [0, 8]]$$
$$MM := [7325, 898, 6338, 777]$$
$$t := 314159265$$
$$t := 271828182$$
$$t := 1$$

```
{--> enter ced, args = 41469, 11580, 4, 1, 0
```

$$s := 1$$
$$nn := 4$$
$$aa := 41469$$
$$bb := 11580$$
$$ep := 1$$
$$np := 2$$
$$f := 256$$
$$ap := 161$$
$$bp := 45$$
$$app := 253$$
$$bpp := 60$$
$$wp := 1$$

```
{--> enter ced, args = 161, 45, 2, 1, 1
```

$$s := 16$$

```
{--> enter sced, args = 161, 45, 16
```

$$L := [161, 45, 1, 0, 0, 1]$$
$$L := [26, 45, 1, 3, 0, 1]$$
$$L := [26, 19, 4, 3, 1, 1]$$
$$L := [7, 19, 4, 3, 1, 1]$$

$$[7, 19, 4, 3, 1, 1]$$

```
<-- exit sced (now in ced) = [7, 19, 4, 3, 1, 1]}
<-- exit ced (now in ced) = [7, 19, 4, 3, 1, 1]}
```

$$L := [7, 19, 4, 3, 1, 1]$$

$$M := [4, 3, 1, 1]$$

$$aa := 7$$

$$bb := 19$$

$$aa := 26$$

$$aa := 6729$$

$$bb := 4851$$

$$t := 41469$$

$$t := 11580$$

$$t := 1$$

$$Q := [\,]$$

$$L := [1878, 4851, 1, 1, 0, 1]$$

$$aa := 1878$$

$$bb := 4851$$

$$Q := [[1, 0]]$$

$$L := [1878, 1095, 1, 0, 2, 1]$$

$$aa := 1878$$

$$bb := 1095$$

$$Q := [[1, 0], [0, 2]]$$

$$MM := [18, 7, 5, 2]$$

$$t := 41469$$

$$t := 11580$$

$$t := 1$$

$$nn := 3$$

$$f := 16$$

$$ap := 117$$

$$bp := 68$$

$$app := 6$$

$$bpp := 7$$

$$wp := 1$$

```
{--> enter ced, args = 117, 68, 2, 1, 1
```

$$s := 16$$

```
{--> enter sced, args = 117, 68, 16
```

$$L := [117, 68, 1, 0, 0, 1]$$

$$L := [49, 68, 1, 1, 0, 1]$$

$$L := [49, 19, 2, 1, 1, 1]$$

$$L := [11, 19, 2, 3, 1, 2]$$

$$[11, 19, 2, 3, 1, 2]$$

```
<-- exit sced (now in ced) = [11, 19, 2, 3, 1, 2]}
<-- exit ced (now in ced) = [11, 19, 2, 3, 1, 2]}
```

$$L := [11, 19, 2, 3, 1, 2]$$

$$MM := [2, 3, 1, 2]$$

$$aa := 11$$

$$bb := 19$$

$$aa := 30$$

$$aa := 471$$

$$bb := 312$$

$$MM := [7, 11, 5, 8]$$

$$M := [43, 68, 12, 19]$$

$$t := 41469$$

$$t := 11580$$

$$t := 1$$

$$Q := [\,]$$

$$L := [159, 312, 1, 1, 0, 1]$$

$$aa := 159$$

$$bb := 312$$

$$Q := [[1, 0]]$$

$$L := [159, 153, 1, 0, 1, 1]$$

$$aa := 159$$

$$bb := 153$$

$$Q := [[1, 0], [0, 1]]$$

$$MM := [154, 111, 43, 31]$$

$$t := 41469$$

$$t := 11580$$

$$t := 1$$

```
{--> enter ced, args = 159, 153, 2, 1, 0
```

$$s := 1$$

```
{--> enter sced, args = 159, 153, 1
```

$$L := [159, 153, 1, 0, 0, 1]$$

$$L := [6, 153, 1, 1, 0, 1]$$

$$L := [6, 3, 26, 1, 25, 1]$$

$$L := [0, 3, 26, 27, 25, 26]$$

$$[0, 3, 26, 27, 25, 26]$$

```
<-- exit sced (now in ced) = [0, 3, 26, 27, 25, 26]}
<-- exit ced (now in ced) = [0, 3, 26, 27, 25, 26]}
```

$$L := [0, 3, 26, 27, 25, 26]$$

$$MM := [26, 27, 25, 26]$$

$$MM := [77, 80, 51, 53]$$

$$M := [6779, 7044, 1893, 1967]$$

$$aa := 0$$

$$bb := 3$$

$$aa := 3$$

$$t := 41469$$

$$t := 11580$$

$$t := 1$$

$$[0, 3, 6779, 7044, 1893, 1967]$$

```
<-- exit ced (now in ced) = [0, 3, 6779, 7044, 1893,
1967]}
```

$$L := [0, 3, 6779, 7044, 1893, 1967]$$

$$MM := [6779, 7044, 1893, 1967]$$

$$MM := [62904, 65363, 56125, 58319]$$

$$M := [51356089, 53363666, 44436163, 46173231]$$

$$aa := 0$$

$$bb := 3$$

$$aa := 3$$

$$t := 314159265$$

$$t := 271828182$$

$$t := 1$$

$$[0, 3, 51356089, 53363666, 44436163, 46173231]$$

```
<-- exit ced (now at top level) = [0, 3, 51356089,
53363666, 44436163, 46173231]}
```

$$[0, 3, 51356089, 53363666, 44436163, 46173231] \quad (7.20.15)$$

```
> 3*(%[3]+%[4]),3*(%[5]+%[6]),%[3]*%[6]-%[4]*%[5];
```

$$314159265, 271828182, 1 \quad (7.20.16)$$

```
> ;
> ;
> # m is the number of clock cycles used to multiple an n-word
  number
  # by a one-word number

  m:='m'; m1:='m1'; m:=proc(n) m1*n end; # m1:=10;
```

$$m := m$$

$$m1 := m1$$

$$m := \mathbf{proc}(n)\ m1 * n\ \mathbf{end\ proc} \quad (7.20.17)$$

```
> m(2);
```

$$2\ m1 \quad (7.20.18)$$

```
> # M is the number of clock cycles used to multiple two n-word
  numbers

  M:='M'; M1:='M1';
  M:=proc(n)
    if n<=4 then n*n*M1
    elif n<=512 then 2*3^log[2](n)*M1
    else 15*n*log[2](n)*M1 fi
  end;
  # M1:=10;
```

$$M := M$$

$$M1 := M1$$

$$M := \mathbf{proc}(n) \quad (7.20.19)$$

$$\mathbf{if}\ n <= 4\ \mathbf{then}$$

$$n * n * M1$$

$$\mathbf{elif}\ n <= 512\ \mathbf{then}$$

$$2 * 3\hat{}\log[2](n) * M1$$

$$\mathbf{else}$$

$$15 * n * \log[2](n) * M1$$

$$\mathbf{end\ if}$$

$$\mathbf{end\ proc}$$

```
> n:=2^13; evalf(log[2](32*n)); evalf(M(n)); evalf(log[2](M(n)/
  (32*n)));
```

$$n := 8192$$

$$18.$$

$$1.597440\,10^{6}\,M1$$

$$1.442695041\,\ln(6.093750000\,M1) \qquad (7.20.20)$$

```
> # a is the number of clock cycles used to calculate sum or
  difference
  # of two n-word numbers

  a:='a'; a1:='a1';
  a:=proc(n) n*a1 end;
  # a1:=6;
```

$$a := a$$

$$a1 := a1$$

$$a := \mathbf{proc}(n)\ n^{*}\,a1\ \mathbf{end\ proc} \qquad (7.20.21)$$

```
> a(1024);
```

$$1024\,a1 \qquad (7.20.22)$$

```
> # r is the number of clock cycles used to calculate remainder
  # by one-word quotient for two n-word numbers

  r:='r'; r1:='r1';
  r:=proc(n) n*r1 end;
  # r1:=a1+m1;
```

$$r := r$$

$$r1 := r1$$

$$r := \mathbf{proc}(n)\ n^{*}\,r1\ \mathbf{end\ proc} \qquad (7.20.23)$$

```
> r(1024);
```

$$1024\,r1 \qquad (7.20.24)$$

```
> # q is the number of clock cycles used to calculate a one-
  word quotient

  q:='q';
  # q:=60;
```

$$q := q \qquad (7.20.25)$$

```
> # This the number of clock cycles used by ced to reduce a 2^n
  word
  # number to the half length

  C:=proc(n) 2*C(n-1)+8*q+4*m(2^n)+4*m(3*2^(n-2))+24*M(2^(n-2))
    +16*a(2^(n-1))+58*c1 end;
```

```
C(0):=300;
C(1):=1000;
# c1:=20  # for call
```

```
C := proc(n)
    2*C(n-1)+8*q+4*m(2^n)+4*m(3*2^(n-2))
    +24*M(2^(n-2))+16*a(2^(n-1))+58*c1
end proc
```

$$C(0) := 300$$

$$C(1) := 1000 \tag{7.20.26}$$

```
> C(2);
```

$$2000 + 8\,q + 28\,m1 + 24\,M1 + 32\,a1 + 58\,c1 \tag{7.20.27}$$

```
> C(3);
```

$$4000 + 24\,q + 112\,m1 + 144\,M1 + 128\,a1 + 174\,c1 \tag{7.20.28}$$

```
> C(4);
```

$$8000 + 56\,q + 336\,m1 + 672\,M1 + 384\,a1 + 406\,c1 \tag{7.20.29}$$

```
> C(5);
```

$$16000 + 120\,q + 896\,m1 + 2640\,M1 + 1024\,a1 + 870\,c1 \tag{7.20.30}$$

```
> C(6);
```

$$32000 + 248\,q + 2240\,m1 + 9168\,M1 + 2560\,a1 + 1798\,c1 \tag{7.20.31}$$

```
> C(7);
```

$$64000 + 504\,q + 5376\,m1 + 30000\,M1 + 6144\,a1 + 3654\,c1 \tag{7.20.32}$$

```
> C(8);
```

$$128000 + 1016\,q + 12544\,m1 + 94992\,M1 + 14336\,a1 + 7366\,c1 \tag{7.20.33}$$

```
> C(9);
```

$$256000 + 2040\,q + 28672\,m1 + 294960\,M1 + 32768\,a1 + 14790\,c1 \tag{7.20.34}$$

```
> C(10);
```

$$512000 + 4088\,q + 64512\,m1 + 904848\,M1 + 73728\,a1 + 29638\,c1 \tag{7.20.35}$$

```
> C(11);
```

$$1024000 + 8184\,q + 143360\,m1 + 2754480\,M1 + 163840\,a1 + 59334\,c1 \tag{7.20.36}$$

```
> C(12);
```

$$2048000 + 16376\,q + 315392\,m1 + 9195360\,M1 + 360448\,a1 + 118726\,c1 \tag{7.20.37}$$

```
> C(13);
```

$$4096000 + 32760\,q + 688128\,m1 + 26500800\,M1 + 786432\,a1 + 237510\,c1 \tag{7.20.38}$$

```
> C(14);
```

$$8192000 + 65528\,q + 1490944\,m1 + 70696320\,M1 + 1703936\,a1 + 475078\,c1 \tag{7.20.39}$$

```
> C(15);
```

$$16384000 + 131064\,q + 3211264\,m1 + 179731200\,M1 + 3670016\,a1 + 950214\,c1 \tag{7.20.40}$$

```
> C(16);
```

$$32768000 + 262136\,q + 6881280\,m1 + 442037760\,M1 + 7864320\,a1 + 1900486\,c1 \tag{7.20.41}$$

```
> C(17);
```

$$65536000 + 524280\,q + 14680064\,m1 + 1061022720\,M1 + 16777216\,a1 + 3801030\,c1 \tag{7.20.42}$$

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

$$3 \tag{7.20.43}$$

```
> ;
>
>
>
>
```

▶ 8. Elliptikus függvények

▶ 9. Számolás elliptikus görbéken

▶ 10. Faktorizálás elliptikus görbékkel

▶ 11. Prímteszt elliptikus görbékkel

▶ 12. Polinomfaktorizálás

▶ 13. Az AKS–teszt

▶ 14. A szita módszerek alapjai

▶ 15. Számtest szita

► 16. Vegyes problémák