POTENČNA ŠTEVILA IN MATHEMATICA

Ko sem v prvi številki dvajsetega letnika Preseka prebral članek *Potenčna števila*, ki ga je napisal *prof. Jože Grasselli*, sem sedel za računalnik, pognal sistem za simbolično računanje Mathematica in pogledal, ali mi lahko računalnik pomaga pri branju tega zanimivega prispevka.

Najprej sem pogledal, kakšno orodje ima Mathematica namenjeno potenčnim številom.

```
Mathematica 2.0 for MS-DOS 386/7 (June 21, 1991)
Copyright 1988-91 Wolfram Research, Inc.

In[1]:= ?Prime*
Prime   PrimePi PrimeQ

In[1]:= ?Power*
Power   PowerExpand PowerMod
```

Nič pametnega ne najdemo pod praštevili *Prime...* niti pod potencami *Power...* Zato se lotimo kar razstavljanja na prafaktorje. Na primer: $100 = 2^2 5^2$ in $99 = 3^2 11^1$. Mathematica razstavlja števila z ukazom *FactorInteger*.

```
In[1]:= FactorInteger[100]
Out[1]= {{2, 2}, {5, 2}}

In[2]:= FactorInteger[99]
Out[2]= {{3, 2}, {11, 1}}
```

Za rezultat dobimo seznam, katerega elementi so zopet dvoelementni seznami. Prvi element je praštevilo, drugi pa ustrezni eksponent.

Pri potenčnih številih so zanimivi predvsem eksponenti. Število 99 ni potenčno, ker se v njegovem razcepu na praštevila pojavlja tudi eksponent 1. Z ukazom *Transpose* v izrazu *Out[2]* zberemo na prvem mestu praštevila, na drugem pa eksponente. Pri tem se izraz *%2* nanaša na *Out[2]*.

```
In[3]:= Transpose[%2]
Out[3]= {{3, 11}, {2, 1}}
```

Z ukazom *Last* dobimo zadnji element seznama, v našem primeru bodo to eksponenti, za tem pa z ukazom *Min* izberemo najmanjšega med njimi.

```
In[4]:= Last[%3]
Out[4]= {2, 1}

In[5]:= Min[%4]
Out[5]= 1
```

Ker je rezultat 1, vemo, da 99 ni potenčno število. Če bi isto zaporedje ukazov izvedli pri številu 100, bi dobili na koncu rezultat 2, ki je več kot 1, in bi upravičeno sklepali, da je število 100 potenčno število.

Na osnovi te ugotovitve lahko zdaj napišemo v Mathematici ukaz z imenom *PotencnoStevilo*, ki za poljubno naravno število pove, ali je potenčno ali ni.

```
In[6]:= PotencnoStevilo[x_] := Min[Last[Transpose[FactorInteger[x]]]] > 1

In[7]:= PotencnoStevilo[10]
Out[7]= False

In[8]:= PotencnoStevilo[1]
Last::nolast: {} has a length of zero and no last element.
Out[8]= Last[{}] > 1

In[9]:= PotencnoStevilo[4]
Out[9]= True
```

Pri preskusu ukaza *PotencnoStevilo* smo ugotovili, da odpove pri argumentu 1. To je razumljivo, saj število 1 ni niti praštevilo niti ni sestavljeno število. Zato ukaz *FactorInteger[1]* vrne prazen seznam, ta pa nima zadnjega elementa.

V naslednji vrstici uporabimo vgrajeni ukaz *Table*. Ukaz *Table* nam vrne seznam. V našem primeru je to seznam parov oblike {*i,PotencnoStevilo[i]*}, pri čemer *i* zavzame zapored vrednosti 1,2, ..., 40.

```
In[10]:= Table[{i,PotencnoStevilo[i]},{i,40}]
Last::nolast: {} has a length of zero and no last element.
Out[10]= {{1, Last[{}] > 1}, {2, False}, {3, False}, {4, True}, {5, False},
> {6, False}, {7, False}, {8, True}, {9, True}, {10, False}, {11, False},
> {12, False}, {13, False}, {14, False}, {15, False}, {16, True},
> {17, False}, {18, False}, {19, False}, {20, False}, {21, False},
> {22, False}, {23, False}, {24, False}, {25, True}, {26, False},
> {27, True}, {28, False}, {29, False}, {30, False}, {31, False},
> {32, True}, {33, False}, {34, False}, {35, False}, {36, True},
> {37, False}, {38, False}, {39, False}, {40, False}}
```

Očitno so v zgornjem seznamu potenčna števila le tista, poleg katerih stoji zapisan *True*. V Mathematici jih lahko izberemo iz seznama z ukazom *Select*. Če pozabimo, kako natančno moramo uporabiti ukaz, lahko povprašamo za nasvet Mathematico.

```
In[11]:= ?Select
Select[list, crit] picks out all elements ei of list for which crit[ei] is
  True. Select[list, crit, n] picks out the first n elements for which
  crit[ei] is True.
```

Še kratek preizkus:

```
In[11]:= Select[Table[i,{i,2,100}],PotencnoStevilo]
Out[11]= {4, 8, 9, 16, 25, 27, 32, 36, 49, 64, 72, 81, 100}
```

in zdaj zares:

```
In[12]:= Select[Table[i,{i,2,10000}],PotencnoStevilo]
Out[12]= {4, 8, 9, 16, 25, 27, 32, 36, 49, 64, 72, 81, 100, 108, 121, 125,
> 128, 144, 169, 196, 200, 216, 225, 243, 256, 288, 289, 324, 343, 361,
> 392, 400, 432, 441, 484, 500, 512, 529, 576, 625, 648, 675, 676, 729,
> 784, 800, 841, 864, 900, 961, 968, 972, 1000, 1024, 1089, 1125, 1152,
> 1156, 1225, 1296, 1323, 1331, 1352, 1369, 1372, 1444, 1521, 1568, 1600,
> 1681, 1728, 1764, 1800, 1849, 1936, 1944, 2000, 2025, 2048, 2116, 2187,
> 2197, 2209, 2304, 2312, 2401, 2500, 2592, 2601, 2700, 2704, 2744, 2809,
> 2888, 2916, 3025, 3087, 3125, 3136, 3200, 3249, 3267, 3364, 3375, 3456,
> 3481, 3528, 3600, 3721, 3844, 3872, 3888, 3969, 4000, 4096, 4225, 4232,
> 4356, 4489, 4500, 4563, 4608, 4624, 4761, 4900, 4913, 5000, 5041, 5184,
> 5292, 5324, 5329, 5400, 5408, 5476, 5488, 5625, 5776, 5832, 5929, 6075,
> 6084, 6125, 6241, 6272, 6400, 6561, 6724, 6728, 6859, 6889, 6912, 7056,
> 7200, 7225, 7396, 7569, 7688, 7744, 7776, 7803, 7921, 8000, 8100, 8192,
> 8281, 8464, 8575, 8649, 8712, 8748, 8788, 8836, 9000, 9025, 9216, 9248,
> 9261, 9409, 9604, 9747, 9800, 9801, 10000}

In[13]:= Length[%12]
Out[13]= 184
```

Med prvimi 10000 naravnimi števili je 184 potenčnih. Pri tem smo si pomagali z vgrajenim ukazom *Length*, ki vrne dolžino seznama.

Kako blizu pa so si sosednja potenčna števila? V članku profesorja Grassellija je dokazano, da je v zaporedju potenčnih števil neskončno parov takih števil, da sta števili para zaporedni naravni števili.

Pri preiskovanju bomo uporabili ukaz *Partition*.

```
In[14]:= ?Partition
Partition[list, n] partitions list into non-overlapping sublists of length n.
  Partition[list, n, d] generates sublists with offset d. Partition[list,
  {n1, n2, ...}, {d1, d2, ...}] partitions successive levels in list into
  length ni sublists with offsets di.
```

Iz zaporedja hočemo napraviti zaporedje sosednjih parov. Kdor ne razume angleško, naj ugane, kako deluje ukaz *Partition* iz naslednjega primera.

```
In[14]:= Partition[Table[i,{i,10}],2,1]
Out[14]= {{1, 2}, {2, 3}, {3, 4}, {4, 5}, {5, 6}, {6, 7}, {7, 8}, {8, 9},
> {9, 10}}
```

Učinek naslednje vrstice lahko razumemo, težje pa je razložiti natančno delovanje vgrajenega ukaza *Apply*. Pri tem *Subtract* seveda pomeni odštevanje.

```
In[15]:= Apply[Subtract,%14,1]
Out[15]= {-1, -1, -1, -1, -1, -1, -1, -1, -1}
```

Prvi element v seznamu *Out[15]* ustreza razliki $1-2$, drugi $2-3$, in tako dalje. Vse skupaj ponovimo na zaporedju potenčnih števil, le da na koncu z vgrajenim ukazom *Minus* zamenjamo predznake števil v seznamu.

```
In[16]:= Partition[%12,2,1]
Out[16]= {{4, 8}, {8, 9}, {9, 16}, {16, 25}, {25, 27}, {27, 32}, {32, 36},
> {36, 49}, {49, 64}, {64, 72}, {72, 81}, {81, 100}, {100, 108},
...
> {9025, 9216}, {9216, 9248}, {9248, 9261}, {9261, 9409}, {9409, 9604},
> {9604, 9747}, {9747, 9800}, {9800, 9801}, {9801, 10000}}
In[17]:= Apply[Subtract,%16,1]
Out[17]= {-4, -1, -7, -9, -2, -5, -4, -13, -15, -8, -9, -19, -8, -13, -4, -3,
> -16, -25, -27, -4, -16, -9, -18, -13, -32, -1, -35, -19, -18, -31, -8,
...
> -32, -27, -118, -79, -100, -92, -89, -183, -111, -74, -63, -36, -40,
> -48, -164, -25, -191, -32, -13, -148, -195, -143, -53, -1, -199}
In[18]:= Minus[%17]
Out[18]= {4, 1, 7, 9, 2, 5, 4, 13, 15, 8, 9, 19, 8, 13, 4, 3, 16, 25, 27, 4,
> 16, 9, 18, 13, 32, 1, 35, 19, 18, 31, 8, 32, 9, 43, 16, 12, 17, 47, 49,
> 23, 27, 1, 53, 55, 16, 41, 23, 36, 61, 7, 4, 28, 24, 65, 36, 27, 4, 69,
> 71, 27, 8, 21, 17, 3, 72, 77, 47, 32, 81, 47, 36, 36, 49, 87, 8, 56, 25,
> 23, 68, 71, 10, 12, 95, 8, 89, 99, 92, 9, 99, 4, 40, 65, 79, 28, 109,
> 62, 38, 11, 64, 49, 18, 97, 11, 81, 25, 47, 72, 121, 123, 28, 16, 81,
> 31, 96, 129, 7, 124, 133, 11, 63, 45, 16, 137, 139, 13, 87, 41, 143,
> 108, 32, 5, 71, 8, 68, 12, 137, 151, 56, 97, 146, 9, 41, 116, 31, 128,
> 161, 163, 4, 131, 30, 23, 144, 144, 25, 171, 173, 119, 56, 32, 27, 118,
> 79, 100, 92, 89, 183, 111, 74, 63, 36, 40, 48, 164, 25, 191, 32, 13,
> 148, 195, 143, 53, 1, 199}
```

Z ukazom *Sort* zaporedje uredimo, z ukazom *Union* pa združimo enake elemente.

```
In[19]:= Sort[%18]
Out[19]= {1, 1, 1, 1, 2, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 5, 5, 7, 7, 7, 8, 8,
> 8, 8, 8, 8, 8, 9, 9, 9, 9, 9, 9, 10, 11, 11, 11, 12, 12, 12, 13, 13, 13,
...
> 137, 139, 143, 143, 144, 144, 146, 148, 151, 161, 163, 164, 171, 173,
> 183, 191, 195, 199}
```

```
In[20]:= Union[%19]
Out[20]= {1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 21, 23,
> 24, 25, 27, 28, 30, 31, 32, 35, 36, 38, 40, 41, 43, 45, 47, 48, 49, 53,
> 55, 56, 61, 62, 63, 64, 65, 68, 69, 71, 72, 74, 77, 79, 81, 87, 89, 92,
> 95, 96, 97, 99, 100, 108, 109, 111, 116, 118, 119, 121, 123, 124, 128,
> 129, 131, 133, 137, 139, 143, 144, 146, 148, 151, 161, 163, 164, 171,
> 173, 183, 191, 195, 199}
```

Iz *Out[18]* vidimo, da so do 10000 štirje zaporedni pari potenčnih števil: (8,9), (288,289), (675,676) in (9800,9801). Z Mathematico zlahka faktoriziramo zadnje tri pare.

```
In[21]:= FactorInteger[288]
Out[21]= {{2, 5}, {3, 2}}

In[22]:= FactorInteger[289]
Out[22]= {{17, 2}}

In[23]:= FactorInteger[675]
Out[23]= {{3, 3}, {5, 2}}

In[24]:= FactorInteger[676]
Out[24]= {{2, 2}, {13, 2}}

In[25]:= FactorInteger[9800]
Out[25]= {{2, 3}, {5, 2}, {7, 2}}

In[26]:= FactorInteger[9801]
Out[26]= {{3, 4}, {11, 2}}
```

Za konec zastavimo bralcu še nekaj vprašanj. V zaporedju razlik *Out[20]* je največje število 199. To razliko doseže en sam par potenčnih števil do 10000, ravno $10000 - 9801 = 199$. V zaporednju *Out[20]*, oziroma v *Out[18]* manjkajo števila 6, 14, 20, 22, ... Ker smo opazovali potenčna števila le do 10000, je razumljivo, da so v zaporednju *Out[20]* luknje. Če bi mejo 10000 porinili navzgor, bi se te luknje gotovo polnile. Ni pa jasno, ali bi se tudi vse zapolnile.

Vprašanje 1. Ali za poljubno naravno število n obstaja tak par sosednjih potenčnih števil p in P, da je $P - p > n$?

Negativen odgovor na to vprašanje bi pomenil tudi negativen odgovor na naslednje vprašanje.

Vprašanje 2. Ali za poljubno naravno število n obstaja tak par sosednjih potenčnih števil p in P, da je $P - p = n$?

Tudi zadnje vprašanje je po svoje zanimivo. Če je odgovor nanj pozitiven, ga morda lahko najdemo z ugibanjem in preskušanjem, torej z Mathematico. Če pa je odgovor negativen, potem brez matematike ne bo šlo.

Vprašanje 3. Poišči najmanjši par sosednjih potenčnih števil p in P, da bo $P - p = 6$, ali dokaži, da takega para ni.

Na srečo je odgovor na prvo vprašanje pozitiven. Na tem mestu tega ne bomo dokazovali. Z Mathematico bomo le poiskali empirične argumente za to trditev. Ideja dokaza je preprosta. Naj $p(k)$ označuje število potenčnih števil, ki niso večja od k, in naj bo $n(k) = \max\{i/(p(i)+1); i \leq k\}$. Ker je prvo potenčno število 4, drugo pa 8, je smiselno opazovati $n(k)$ za vrednosti $k \geq 8$. Zaradi lažjega računanja bomo privzeli, da sta tudi 0 in k potenčni števili. Tedaj med prvimi k naravnimi števili obstaja vsaj en par sosednjih potenčnih števil P in p, tako da je $P - p \geq n(k)$. Lahko bi se sicer zgodilo, da je $P = k$ in k ni potenčno število, vendar tedaj gledamo na k kot na zamenjavo za naslednje večje potenčno število. Taka zamenjava ne pokvari ocene za sosednji potenčni števili. Z Mathematico si lahko narišemo, kako naraščajo vrednosti $n(k)$. Z $r(k)$ bomo označili maksimalno razliko med dvema sosednjima potenčnima številoma, od katerih nobeno ni večje od k. Tudi tu sta 0 in k upoštevani kot potenčni števili.

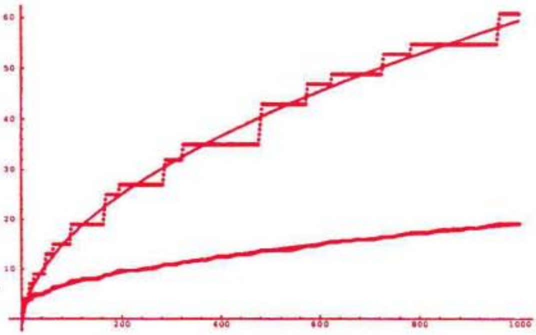
Program je rekurziven in uporablja zahtevne objektno orientirane zmogljivosti Mathematice, zato ga na tem mestu ne bomo podrobneje razlagali:

```
n[0] = 0
n[k_] := n[k] = If[k > (p[k] + 1)*n[k - 1], N[k/(p[k] + 1)], n[k - 1]]
p[1] = 0
p[k_] := p[k] = If[PotencnoStevilo[k], 1 + p[k - 1], p[k - 1]]
r[0] = 0
r[k_] := r[k] = If[p[k] == 0, k, If[k < b[k] + r[b[k]], r[b[k]], k - b[k]]]
b[k_] := b[k] = If[p[k] == 0, 0, If[PotencnoStevilo[k - 1], k - 1, b[k - 1]]]
```

Rezultate do 1000 dobimo najprej v obliki seznamov:

```
In[27]:= Table[n[k],{k,1000}]
Out[27]= {1., 2., 3., 3., 3., 3., 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5,
> 3.75, 3.75, 3.75, 3.75, 3.8, 4., 4.2, 4.4, 4.6, 4.8, 4.8, 4.8, 4.8, 4.8,
...
> 19.2, 19.2, 19.2, 19.2, 19.2, 19.2, 19.2, 19.2, 19.2, 19.2, 19.2, 19.2,
> 19.2, 19.2, 19.2, 19.2, 19.2, 19.2}

In[28]:= Table[r[k],{k,1000}]
Out[28]= {1, 2, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 6, 7, 7, 7, 7, 7, 7, 7,
> 7, 8, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 10,
...
> 61, 61, 61, 61, 61, 61, 61, 61, 61, 61, 61, 61, 61, 61, 61, 61, 61, 61,
> 61, 61, 61, 61, 61, 61, 61, 61, 61, 61, 61, 61, 61, 61, 61, 61, 61, 61}
```



Slika 1: Funkciji $n(k)$ in $r(k)$, za $k = 1, 2, ..., 1000$, hkrati z njunima aproksimacijama $\nu(k) = 1.48473 + 0.557407\sqrt{k}$ in $\rho(k) = -2.94041 + 1.98026\sqrt{k}$.

V obeh primerih števila počasi naraščajo. Z ukazom *ListPlot* bi lahko tabeli narisali. Oblika krivulje obakrat spominja na funkcijo $f(x) = \sqrt{x}$. Ukaz *Fit* je v Mathematici namenjen aproksimaciji podatkov. Z njim bomo poskušali poiskati najboljše vrednosti za a in b, tako da bodo podatki kar najmanj odstopali od funkcije $a + b\sqrt{x}$.

```
In[29]:= ?Fit*
Fit[data, funs, vars] finds a least-squares fit to a list of data as a linear
  combination of the functions funs of variables vars. The data can have the
  form {{x1, y1, ..., f1}, {x2, y2, ..., f2}, ...}, where the number of
  coordinates x, y, ... is equal to the number of variables in the list vars.
  The data can also be of the form {f1, f2, ...}, with a single coordinate
  assumed to take values 1, 2, .... The argument funs can be any list of
  functions that depend only on the objects vars.

In[29]:= Fit[%27,{1,Sqrt[x]},x]
Out[29]= 1.48473 + 0.557407 Sqrt[x]

In[30]:= Fit[%28,{1,Sqrt[x]},x]
Out[30]= -2.94041 + 1.98026 Sqrt[x]
```

Za $n(k)$ smo dobili torej $a = 1.48473$ in $b = 0.557407$, za $r(k)$ pa smo dobili $a = -2.94041$ in $b = 1.98026$. Na sliki 1 vidimo ob podatkih iz *Out[27]* in *Out[28]* še funkciji iz *Out[29]* in *Out[30]*. Našli smo dovolj indicev za hipotezo, da je med prvimi n naravnimi števili kvečjemu $c\sqrt{n}$ potenčnih števil, pri čemer je c izbrana konstanta. Ker so vsi kvadrati potenčna števila, konstanta c ne more biti manjša od 1. Naši eksperimenti kažejo, da bi prava vrednost utegnila biti okoli števila 2. Strog dokaz te hipoteze pa prepuščamo bralcu.

Tomaž Pisanski