

整数の《平方値のキャラクターの個数》による分類

統計数理研究所(名誉教授) 鈴木義一郎

([:~."@*:] 301 9061 [:~."@*:] 305 93025	('0'-.~[:~."@*:] 301 961 '0'-.~[:~."@*:] 305 9325	「.:」は文字化する関数 「~.」は重複要素の排除 「-.」は集合間での引算
--	--	--

no0=[:#@~."@*:] no0 301 4 no0 305 5 no0"0(301 305) 4 5 nub0=:4 :0 k=.x-1 [ r=.'' while.k<y do.r=r,no0 k=.k+1 end. )	no=[:#@'0'-.~[:~."@*:] no 301 3 no 305 4 no"0(301 305) 3 4 nub=:4 :0 k=.x-1 [ r=.'' while.k<y do.r=r,no k=.k+1 end. )	「no0」は'0'まで含めて要素の個数を出力する関数で、「no」は'0'を除いて要素の個数を出力する関数である。  「nub0」は左引数(x)から右引数(y)までの整数値に対する平方値のキャラクターの個数を出力する。  「nub」は'0'を除いた場合の関数である。
301 nub0 305 4 5 4 5 5	301 nub 305 3 4 3 5 4	*:301+i.5 90601 91204 91809 92416 93025
freq0=:(i.10),[:+/(i.10)=/~nub0 301 freq0 305 0 1 2 3 4 5 6 7 8 9 0 0 0 0 2 3 0 0 0 0	freq=:(i.10),[:+/(i.10)=/~nub 301 freq 305 0 1 2 3 4 5 6 7 8 9 0 0 0 2 2 1 0 0 0 0	+/{:F 290
]F=:1 freq 290 0 1 2 3 4 5 6 7 8 9 0 8 40 110 94 38 0 0 0 0	set2=:4 :0 k=.x-1 [ r=.'' while.k<y do.if.2=no k=.k+1 do.r=r,k	5 8 \$ 1 set2 290 4 5 6 7 8 9 11 12 15 21 22 26 38 40 45 47 50 60 70 80 88 90 97 101

```

end.
end.
)
102 109 110 120 149 150 173 201
202 210 212 220 235 260 264 283

```

<pre> 291 freq 4000 0 1 2 3 4 5 6 7 8 9 0 4 56 432 1121 1388 629 77 3 0 </pre>	<pre> 8 7 \$ 291 set2 4000 298 380 400 450 470 500 505 547 548 600 638 700 745 800 813 880 900 901 970 997 1001 1002 1009 1010 1011 1020 1049 1090 </pre>
<pre> _8{"1(4001 freq 15000) 2 3 4 5 6 7 8 9 50 447 2050 4033 3359 979 77 4 </pre>	<pre> 1100 1101 1200 1415 1485 1490 1498 1500 1703 1730 2001 2002 2010 2020 2022 2100 2120 2200 2202 2350 2600 </pre>
<pre> 2{"1(15001 freq 200000) 2 35 </pre>	<pre> 2640 2830 2980 3114 3148 3800 4000 </pre>
<pre> _8{"1(15001 freq 200000) 2 3 4 5 6 7 8 9 35 569 4717 23595 60446 66768 26167 2701 </pre>	
<pre> 7 5 \$ S=:15001 set2 200000 15985 17030 17300 20001 20002 20010 20020 20022 20100 20102 20200 20220 20348 20602 20992 21000 21200 21602 22000 22002 22020 23500 26000 26400 27749 28300 28998 29800 31140 31480 38000 40000 44950 45000 100000 </pre>	<p>数値の桁数が多くなると、平方値が指数表示になってしまうので、かなりの部分が抜け落ちてしまっている！</p>

「NO」は「no」という関数を倍精度で平方値を算出するように修正した関数である。	
<pre>NO=:3 :'#("0)-.~.":x*:y'  NUB=:4 :0 k=.x-1 [ r=' ' while.k&lt;y   do.r=.r,NO k=.k+1 end. )</pre>	<pre>FREQ=: (i.10),[:+/(i.10)=/~NUB  SET2=:4 :0 k=.x-1 [ r=' ' while.k&lt;y   do.if.2=NO k=.k+1 do.r=.r,k end. end. )</pre>

<pre>1 FREQ 290 0 1 2 3 4 5 6 7 8 9 0 8 40 110 94 38 0 0 0 0 291 FREQ 4000 0 1 2 3 4 5 6 7 8 9 0 4 56 432 1121 1388 629 77 3 0 15001 FREQ 200000</pre>	<pre>4001 FREQ 15000 0 1 2 3 4 5 6 7 8 9 0 1 50 447 2050 4033 3359 979 77 4  (ここまでは、「freq」という関数の出力結果と一致している。)</pre>	
<pre>0 1 2 3 4 5 6 7 8 9 0 4 102 1557 12709 45652 68259 45389 10485 843</pre>		
<pre>\$ T=:15001 SET2 200000</pre>	<pre>{:T 173000</pre>	<pre>x*:173000 29929000000</pre>
<pre>102</pre>		

```
6 17 $ T
15985 17030 17300 20001 20002 20010 20020 20022 20100
20102 20200 20220 20348 20602 20992 21000 21200
21602 22000 22002 22020 23500 26000 26400 27749 28300
28998 29800 31140 31480 38000 40000 44950 45000
47000 50000 50005 50050 50500 54700 54800 60000 63800
64980 70000 72495 74499 74500 80000 81300 81619
82260 88000 90000 90001 90010 90100 97000 99700 99970
99997 100001 100002 100009 100010 100011 100020 100090
100100 100101 100110 100200 100549 100900 100901 101000 101001
101100 102000 103010 104010 104900 104960 106790 108010
```

109000 110000 110001 110010 110100 120000 124499 141500 144990  
148500 148998 149000 149800 150000 159850 170300 173000