下呂数

田中 二郎[1,a)]

概要：下呂数は 2015 年の夏のプログラミング・シンポジウムで竹内先生が出題されたものである．単純な定義ながらも, その性質は不明な点が多く, 計算機の課題としても興味深い．特に, ある長さ i の下呂数のうち, 最大なものは 3^i であるのに対し, 最小の数を求める良い方法は 見つけられていない．しかし, おおよその値が $2 \times 10^{(i-1)/4}$ であることが, 長さ 43 までの下呂数から予想される．
詳細は http://gakkan.net/jiro/whoami/gero/ にて公開している．識者の意見を求む.

キーワード：プログラミング・シンポジウム，冬，予稿集, 下呂数

1. はじめに

下呂数は 2015 年の夏のプログラミング・シンポジウムで竹内先生が出題されたものである.[*1]

- ０からはじめて, ＋２, ＋３, ∗２, ∗３の演算を左から順に実行する.
- そのうち, 最短のものを下呂数という

例）

2 ＝ 0+2
3 ＝ 0+3
4 ＝ 0+2+2，0+2*2
5 ＝ 0+3+2，0+2+3
6 ＝ 0+3+3，0+2*3，0+3*2
（0+2+2+2 は, 長いのでダメ）
7 ＝ 0+3+2+2，0+2+3+2，0+2+2+3，0+2*2+3，
8 ＝ 0+3+3+2，0+2*3+2，0+3*2+2，0+3+2+3，
0+2+3+3，0+2+2*2，0+2*2*2
9 ＝ 0+3*3

2. 拡張下呂数

ある自然数 n について, 拡張下呂数 $g(n)$ を考える. 先頭の「０」は表示しない.

$g(1) = \{\}$
$g(2) = \{+2\}$
$g(3) = \{+3\}$
$g(n) = g(n-2) + 2$ if $n > 3$
or $g(n-3) + 3$ if $n > 4$
or $g(n/2) * 2$ if $n > 3$ and $n \bmod 2 = 0$
or $g(n/3) * 3$ if $n > 5$ and $n \bmod 3 = 0$

例）

g(6)= g(4) ＋２,g(3) ＋３,g(3) ∗２,g(2) ∗３
= { +2+2+2, +2*2+2, +3+3, +3*2, +2*3 }

位数（長さ）を限定した拡張下呂数 $g(n, i)$ を考える.

g(6, 2) = { +3+3, +3*2, +2*3 }
g(6, 3) = { +2+2+2, +2*2+2 }

$g(n, i)$ で, 空でない最小の i を $g_i(n)$ とする.

$g(n, g_i(n))$ の要素の個数を $g_c(n)$ とする.

1 開智国際大学
a) jiro@kaichi.ac.jp
*1 http://cybozushiki.cybozu.co.jp/articles/m000439.html

n	$g_i(n)$	$g_c(n)$	$g(n)$
1	-	0	
2	1	1	+2
3	1	1	+3
4	2	2	+2+2, +2*2
5	2	2	+3+2, +2+3
6	2	3	+3+3, +3*2, +2*3 +2+2+2, +2*2+2
7	3	4	+3+2+2, +2+3+2, +2+2+3, +2*2+3
8	3	7	+3+3+2, +3*2+2, +2*3+2, +3+2+3, +2+3+3, +2+2*2, +2*2*2 +2+2+2+2, +2*2+2+2
9	2	1	+3*3 +3+3+3, +3*2+3, +2*3+3 +3+2+2+2, +2+3+2+2, +2+2+3+2, +2*2+3+2, +2+2+2+3, +2*2+2+3

3. 位数と最少下呂数, 最大下呂数

ある位数 i である下呂数のうち, 最少のものを最少下呂数 $g_{min}(i)$, 最大のものを最大下呂数 $g_{max}(i)$ とすると, 最大下呂数 $g_{max}(i) = 3^i$ であるが, 最少下呂数 $g_{min}(i)$ は式であらわすことはできない.

i	$g_{min}(i)$	$g_{max}(i)$
1	2	3
2	4	9
3	7	27
4	13	81
5	19	243
6	37	729
7	55	2187
8	109	6561
9	199	19683
10	325	59049
11	631	177147
12	973	531441
13	1943	1594323
14	3349	4782969
15	6157	14348907
16	10045	43046721
17	19441	129140163
18	34993	387420489
19	69983	1162261467

i	$g_{min}(i)$
20	108865
21	213841
22	326593
23	641521
24	1259713
25	2239489
26	3895777
27	6718465
28	12317185
29	20155393
30	36951553
31	60466177
32	120932351
33	181398529
34	362797055
35	725594077
36	1503256321
37	2176782337
38	4642458625
39	6530347009
40	15237476353
41	27088846849
42	53935828993
43	81266540545

ところで, $g_i(3^i) = i$ であるが, $g_i(2^i) \neq i$ である.

e.g. $g(8192) = +3*3+2*3*3+2*3*3*3+3*3+2$

$g_{min}(i)$ は, およそ $2*10^{(i-1)/4}$ であることが推定されるが, これは定義から導き出されたものではなく, 134,326,599,670 までの下呂数を計算した結果から類推したものである.

また, $g_{min}(i)$ は 6 の倍数 + 1 に思えるが, $g_i(1943) = 13$, $g_i(69983) = 19$, $g_i(120932351) = 32$, $g_i(362797055) = 34$ という反証がある.

4. 下呂数の個数 $g_c(n)$

$g_c(n)$ について, 当初 $g_c(n)/n < 0.4$ と予想したが, n が大きくなるにつれて $g_c(n)/n$ は小さくなった. 小さくなる割合は, n が 2 倍になるごとに, およそ 0.9 倍であるが, 明確な関係があるようには思われない.

5. プログラム例

以下に, 単純に $g_i(n)$ を計算するプログラムの例を示す.

```
#include <stdio.h>
#include <stdlib.h>
#define min(x,y)    (((x) < (y)) ? (x) : (y))
#define SIZE    140703107645440

int main(int argc, char *argv[])
{
    unsigned char *g = malloc(SIZE);
    g[2] = 1;
    g[3] = 1;
    g[4] = 2;
    printf("n\tgi(n)\n2\t1\n3\t1\n4\t2\n");
    for(unsigned long n = 5; n < SIZE; n ++) {
        unsigned char x = min(g[n-2], g[n-3]);
        if((n % 2) == 0) x = min(x, g[n/2]);
        if((n % 3) == 0) x = min(x, g[n/3]);
        g[n] = ++ x;
        printf("%ld\t%d\n", n, x);
    }
}
```

環境は FreeBSD10.2, swap は 256GB, SIZE は Segmentation falt の出ない最大値で, 134,326,599,670 まで計算できた.

6. まとめ

下呂数の計算は $min()$ を使う都合上, 最適化できないと思われる. OS のメモリ管理を利用する単純なプログラムでは記憶容量を生かすことはできなかったが, ファイルを記憶域として使用すれば, ファイルシステムの限界まで計算できるであろう.

ぜひとも, $g_{min}(44)$ 以降を計算したい.

また, $g_c(n)$ の分布や, $g_c(n) = 1$ となる n の分布などについても調べてみたいと思っている.

謝辞 下呂数を出題された竹内先生に感謝します.