

分割数 (partition number)

Masato Shimura
JCD02773@nifty.ne.jp

2008年4月22日

概要

分割数を積み上げ方式で機械的に数え上げる計算方式のスク립トを作成した。

1 分割数

自然数の分割： $n = \lambda_1 + \lambda_2 + \dots + \lambda_r$

$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r$ と並べる。

次にスク립ト作成のための何らかの法則を見つける。

1. 積み上げは 2,3,4,5,...,n
2. 2 の後ろに付く 1 の数は 2 行前、同じく 3 に付く数 1 の数は 3 行前の 1 の項 ... と組み合わせることが出来る。
3. 前後に ((#y)#1) と y を入れる。(e.g. 11111 と 5)
4. 分割した数にはその数を越える数は付加しない。

1.1 積み上げ方式

分割数 (partition number) のカウントを積み上げ方式で考える。(効率は無視し、頭のこんがらかる再帰は用いない。)

```
+-----+-----+-----+-----+-----+-----+-----+-----+
|1          |          |          |          |          |          |          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|1 1        |2          |          |          |          |          |          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|1 1 1      |2 1        |3          |          |          |          |          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|1 1 1 1    |2 1 1      |2 2        |3 1  |4          |          |          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|1 1 1 1 1  |2 1 1 1    |2 2 1      |3 1 1|3 2      |4 1  |5          |          |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

```
|1 1 1 1 1 1|2 1 1 1 1|2 2 1 1|2 2 2|3 1 1 1|3 2 1|3 3|4 1 1|4 2|5 1|6|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

1.2 結果

5 3 \$ |. partition 7

```
+-----+-----+-----+
|7          |6 1          |5 2          |
+-----+-----+-----+
|5 1 1      |4 3          |4 2 1        |
+-----+-----+-----+
|4 1 1 1    |3 3 1        |3 2 2        |
+-----+-----+-----+
|3 2 1 1    |3 1 1 1 1    |2 2 2 1      |
+-----+-----+-----+
|2 2 1 1 1|2 1 1 1 1 1|1 1 1 1 1 1|
+-----+-----+-----+
```

partition_nr0 10

```
+---+---+---+---+---+---+---+---+---+---+
|1 1| | | | | | | | |
+---+---+---+---+---+---+---+---+
|1 1|2 1| | | | | | | |
+---+---+---+---+---+---+---+---+
|1 1|2 1|3 1| | | | | | |
+---+---+---+---+---+---+---+---+
|1 1|2 2|3 1|4 1| | | | | |
+---+---+---+---+---+---+---+---+
|1 1|2 2|3 2|4 1|5 1| | | | |
+---+---+---+---+---+---+---+---+
|1 1|2 3|3 3|4 2|5 1|6 1| | | |
+---+---+---+---+---+---+---+---+
|1 1|2 3|3 4|4 3|5 2|6 1|7 1| | |
+---+---+---+---+---+---+---+---+
|1 1|2 4|3 5|4 5|5 3|6 2|7 1|8 1| |
+---+---+---+---+---+---+---+---+
|1 1|2 4|3 7|4 6|5 5|6 3|7 2|8 1|9 1| |
+---+---+---+---+---+---+---+---+
|1 1|2 5|3 8|4 9|5 7|6 5|7 3|8 2|9 1|10 1|
+---+---+---+---+---+---+---+---+
```

兎に角多量のメモリを要するので、100 を越えると顎がでてくる。

```
tc=: 6!:2 NB. tc 'a=. tartition 30'
```

```
tc 'partition 50'
```

```
26.536
```

```
tc 'partition M. 50'
```

```
26.5455
```

再帰や tacit を使っていないので M. は効いていない。

これ以上になると法則を見つけて組み合わせの数だけ計算するほうが良さそうだ。

```
<("2) 5 10 $ partition_nr 50
```

```
+-----+-----+-----+-----+-----+
| 1  1|11  56|21  792|31  6842|41  44583|
| 2  2|12  77|22 1002|32  8349|42  53174|
| 3  3|13 101|23 1255|33 10143|43  63261|
| 4  5|14 135|24 1575|34 12310|44  75175|
| 5  7|15 176|25 1958|35 14883|45  89134|
| 6 11|16 231|26 2436|36 17977|46 105558|
| 7 15|17 297|27 3010|37 21637|47 124754|
| 8 22|18 385|28 3718|38 26015|48 147273|
| 9 30|19 490|29 4565|39 31185|49 173525|
|10 42|20 627|30 5604|40 37338|50 204226|
+-----+-----+-----+-----+-----+
```

ATT の次のページは数列を入力すると何の数列かを答えてくれる。

<http://www.research.att.com/~njas/sequences/index.html>

1, 1, 2, 3, 5, 7, 11, 15, 22, 30, 42, 56, 77, 101, 135, 176, 231, 297, 385, 490, 627, 792, 1002, 1255, 1575, 1958, 2436, 3010, 3718, 4565, 5604, 6842, 8349, 10143, 12310, 14883, 17977, 21637, 26015, 31185, 37338, 44583, 53174, 63261, 75175, 89134 (list; graph; listen)

*1

2 Script

```
partition=: 3 : 0
```

```
NB. partition number
```

*1 表記法は 10 で 42 とする方法と 2 通りあるようである。

```

NB. Usage: e.g. partition 10
ANS=. (<<1),(<1 1;2) NB. 0,1
NR0=: }. >: i. y    NB. first is
NB. -----
for_ctr.  i. <:# NR0 do.  NB. number exam
  TMP2=. ((>:ctr) {. NR0) calc_pm_sub ANS
  ANS=. ANS,< (<tmp # 1),(TMP2),< tmp=.>: ctr { NR0
end.
|. {: ;("1),. ANS
)

calc_pm_sub=: 4 : 0
RW=: |. i. # NR=: x NB. RAW for pickup e.g. 5->2 3 4
ANS2=. <'
  for_ctr.  i. # NR    do. NB. partial of number exam
    Y0=. (ctr{RW) { y
    if. 0=*/ (ctr{NR) check_major Y0
      do. Y0=. ((ctr{NR) check_major Y0) # L:1 Y0 end.
    TMP1=. >(< ctr{NR) , L:0 Y0 NB. connect with ,
    ANS2=. ANS2,<TMP1
  end.
TMP=. ;}. ANS2
)

check_major=: 4 : ' ;*/(L:0) x >:/ L:0 ; y '

```