

数の分割 (Partition) : FINAL VIRSION

帝京平成大学 鈴木義一郎

ここに、4個のみかんがあるとしよう。これを何人かで分けあって食べるとしたら、幾通りの方法が考えられるだろうか。但し、1個のみかんを半分、三分の一、四半分といったように小刻みに分割したりすることは無しとして考える。ここで、何故「みかん」なのかであるが、「ポットン・マンゴー」でも「ピーナッツ」でも構わない。ただマンゴーでは高価すぎるし、ピーナッツでは小さすぎるから、みかんくらいを念頭において考えるのが手頃だろう。要するに、あるモノの分割の仕方、つまり、「整数論」の問題である。

まず、①1人で4個全部を食べるという方法がある。次に2人で分けて食べるとしたら、②3個と1個か、③2個ずつ食べるという2つの方法が考えられる。さらに3人で分けて食べる場合には、④1人だけが2個で残る2人が1個ずつ食べるという方法である最後に、⑤4人で1個ずつ食べる方法である。結局、5通りの方法ということになる。

みかんというより、あるモノが4個ある場合の答えを

4	3 1	2 2	2 1 1	1 1 1 1
---	-----	-----	-------	---------

のように表すことにする。

【「part」という関数を次のように定義する】

<pre>dev=:3 :0 r=.y;<"1(y-t),t=>.i.y-h=<.-:ywhile.h>1 do.r=.r,<(s>0)#s=.s,y-+/s=.(<.y%h)\$h=.h-1 end.)</pre>		<pre>next=:3 0 m=.>:#(1=s)#s=>y h=.((#s)-m){s [t=.(-m){s=>y <h,+/&>(-<:{t)<\(+/t)\$1)</pre>
<pre>devide=:3 :0 t=.r=.y while.(+/.s)>#(s=1)#s=>t do.r=.r,t=.next t end.)</pre>	<pre>part=:3 :0 r=.2{s [t=.(-#s)+<:k=._1){s=.dev y while.k<y-3 do.r=.r,devide(k=.k+1){ t end.)</pre>	
<pre>dev 4 4 3 1 2 2 1 1 1 1 2 2 の次には</pre>	<pre>Next 2 2 2 1 1 が付 加される。</pre>	<pre>part 4 4 3 1 2 2 2 1 1 1 1 1 1 </pre>

【関数「dev」はヘッドの数値を斬減させつつ、スタートのパターンをボックス内に出力】

```
2 5 $ dev 10
```

10	9 1	8 2	7 3	6 4
5 5	4 4 2	3 3 3 1	2 2 2 2 2	1 1 1 1 1 1 1 1 1 1

next<5 5 5 4 1 next<5 4 1 5 3 2	next<5 3 2 5 3 1 1 next<5 3 1 1 5 2 2 1	n ext<5 2 2 1 5 2 1 1 1 n ext<5 2 1 1 1 5 1 1 1 1 1	5 5 から先頭の5を固定し たまま「next」という補助関数を用いて、順次分割のパターンを出力することができる。
--	--	--	--

```
devide<5 5
```

5 5	5 4 1	5 3 2	5 3 1 1	5 2 2 1	5 2 1 1 1	5 1 1 1 1 1
-----	-------	-------	---------	---------	-----------	-------------

上記一連の結果をまとめて出力するのが「devide」という補助関数である。

「part」という関数が正しい結果を出力していることは、以下の結果からも確認できる。

```
10 4$p10=:part 10
```

10	9 1	8 2	8 1 1
7 3	7 2 1	7 1 1 1	6 4
6 3 1	6 2 2	6 2 1 1	6 1 1 1 1
5 5	5 4 1	5 3 2	5 3 1 1
5 2 2 1	5 2 1 1 1	5 1 1 1 1 1	4 4 2
4 4 1 1	4 3 3	4 3 2 1	4 3 1 1 1
4 2 2 2	4 2 2 1 1	4 2 1 1 1 1	4 1 1 1 1 1 1
3 3 3 1	3 3 2 2	3 3 2 1 1	3 3 1 1 1 1
3 2 2 2 1	3 2 2 1 1 1	3 2 1 1 1 1 1	3 1 1 1 1 1 1 1
2 2 2 2 2	2 2 2 2 1 1	2 2 2 1 1 1 1	2 2 1 1 1 1 1 1 1

```
2{.p10
```

2 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1
---------------------	---------------------

```
$ p11=:part 11
```

56

```
2{.p11
```

2 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1
---------------------	---------------------

```
$ p12=:part 12
```

77

```
2{.p12
```

2 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1
	1

【関数【part】の演算時間】

「J6.02」の新機能「M.」を使用しても、その効果は発揮できなかった。NB. tc=:6!:2		
tc' P10=:part 10' 0.0101714 # P10 42	tc' P20=:part 20' 0.110142 # P20 627	tc' P30=:part 30' 1.41232 # P30 5604
tc' P40=:part 40' 32.2255 # P40 37338	tc' P50=:part 50' 1719.47 # P50 204226	分割のパターンの「数」は 急激に増大する！ 時間の方もかかり過ぎて、 とてもヤッテラレナイ！

【ボックスの表示を早めに消去して、数の分割の「個数」だけを導出するプログラム】

「part」を「partno」に変えると、演算時間も大幅に短縮される。		
devno=:3 :0 r=.1 [t=.y while.(+/.s)>#(s=1)#s=.>t do.r=.r+#t=.next t end.)	partno=:3 :0 t=.(-#s)-(r=.3)+k=. _1){.s=.dev y while.k<y-4 do.r=.r+devno(k=.k+1){t end.)	
tc' n40=:partno 40' 4.80605 n40 37338	tc' n50=:partno 50' 25.9093 n50 204226	tc' n60=:partno 60' 123.875 n60 966467
tc' n70=:partno 70' 609.393 n70 4087968	tc' n80=:partno 80' 2567.82 n80 15796476	24 60 60 #: 31109 8 38 29 分割の「個数」だけを求める よう変更した関数にしても、 n=100 の場合で約9時間も 要した！
tc' n90=:partno 90' 9550.78 n90 56634173	tc' n100=:partno 100' 31109 n100 190569292	

<pre>partno=:3 :0 t=. (-#s)-(r=. 3)+k=. _1) {. s=. dev y while. k<y-4 do. r=. r+devno(k=. k+1) {t end. if. y<4 do. 0>. y else. r end.)</pre>	<pre>test0=:3 :0 r=. +/partno"0>(k=. 0) {n=. no y while. k<<:#n do. r=. r, +/partno"0(0>.>(k=. k+1) {n end. -/r)</pre>
--	--

(partno, test0) 12 77 76	(partno, test0) 22 1002 1003	(partno, test0) 35 14883 14882
(partno, test0) 15 176 175	(partno, test0) 26 2436 2437	(partno, test0) 40 37338 37337

part 3

3	2	1	1	1	1
---	---	---	---	---	---

 $p(3) = 3$

part 4

4	3	1	2	2	2	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---

 $p(4) = 5$

part 5

5	4	1	3	2	3	1	1	2	2	1	2	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 $p(5) = 7$

【漸化関係が成立していない！】

$p(5) = _1 + p(4) + p(3) = _1 + 5 + 3 = 7$
$p(7) = _1 + p(6) + p(5) - p(2) = _1 + 11 + 7 - 2 = _1 + 16 = 15$
$p(12) = 1 + p(11) + p(10) - \{p(7) + p(5)\} = (56 + 42) - (15 + 7) = 1 + 76 = 77$
$p(15) = 1 + \{p(14) + p(13)\} - \{p(10) + p(8)\} + p(3) = 135 + 101 - (42 + 22) + 3 = 176$
$p(22) = _1 + \{p(21) + p(20)\} - \{p(17) + p(15)\} + \{p(10) + p(7)\}$ $= _1 + \{792 + 627\} - \{297 + 176\} + \{42 + 15\} = _1 + \{1419\} - \{473\} + \{57\} = 1002$
$p(26) = _1 + \{p(25) + p(24)\} - \{p(21) + p(19)\} + \{p(14) + p(11)\} - p(4)$ $= _1 + \{1958 + 1575\} - \{792 + 490\} + \{135 + 56\} - 5 = _1 + 3533 - 1282 + 191 - 5 = 2436$
$p(35) = 1 + \{p(34) + p(33)\} - \{p(30) + p(28)\} + \{p(23) + p(25)\} - \{p(13) + p(19)\}$ $= 1 + \{12310 + 10143\} - \{5604 + 3718\} + \{1255 + 627\} - \{101 + 30\} = 14883$
$p(40) = 1 + \{p(39) + p(38)\} - \{p(35) + p(33)\} + \{p(28) + p(25)\} - \{p(18) + p(14)\} + p(5)$ $= 1 + \{31185 + 26015\} - \{14883 + 10143\} + \{3718 + 1958\} - \{385 + 135\} + 7 = 37338$

102-L:0 no 102

1 2	5 7	12 15	22 26	35 40	51 57	70 77	92 100
-----	-----	-------	-------	-------	-------	-------	--------

【漸化関係が成立せず、修正を余儀なくされる箇所】

【「test0」にしてみても演算時間は増大し、しかも結果の異なる箇所がある。】

tc'n040=:test0 40' 11.682 tc'n40=:partno 40' 4.947 n040, n40 37337 37338 (値が異なる!)	tc'n050=:test0 50' 66.9439 tc'n50=:partno 50' 26.8823 n050, n50 204226 204226	tc'n060=:test0 60' 350.662 tc'n60=:partno 60' 137.179 n060, n60 966467 966467
--	--	--

【以下に示すように、漸化的に求める方法が結局はベストであると判明した】

step=:3 :0 r10=.1 2 3 5 7 11 15 22 30 42 r10, (+/(<:s) {r10})+/(<:t) {r10['s t'=red y)	tc'r11=:step 11' 0.00529285 r11 1 2 3 5 7 11 15 22 30 42 56
---	--

各ステップでの演算時間に関しては、ほとんど“negregeble”である。

r10 1 2 3 5 7 11 15 22 30 42]s t'=red 11]pp=:(<:s) {r10 42 30]p=:+/pp 72]qq=:(<:t) {r10 11 5]q=:+/qq 16	p-q 56 「p11」の値
{:r11=:r10, p-q 56 's t'=red 12]pp=:(<:s) {r11 56 42]p=:+/pp 98]qq=:(<:t) {r11 15 7]q=:+/qq 22	p-q 76 1+p-q 77 「p12」の値

「r12」には、「r11」に「1+p-q」を付加する。以下の「r15」の場合も同様である。

{:r12=:r11, 1+p-q 77 's t'=red 13]p=:+/(<:s) {r12 134]q=:+/(<:t) {r12 33	p-q 101 「p13」の値
{:r13=:r12, p-q 101 's t'=red 14]p=:+/(<:s) {r13 180]q=:+/(<:t) {r13 45	p-q 135 「p14」の値
{:r14=:r13, p-q]p=:+/(<:s) {r14]q=:+/(<:t) {r14	1+p-q

135 's t':=red 15	239	64	176 「p15」の値
{:r15=:r14, 1+p-q]p=:+/(<:s) {r15]q=:+/(<:t) {r15	p-q
176 's t':=red 16	317	86	231 「p16」の値

【散々悪戦苦闘した結果、以下の4つのプログラムだけで瞬時に算出できることが判明】

no=:3 :0 r=. <1, 1+s=. (k=. 0) {t=. +:&. >:i. y while. y>s do. r=. r, <s, (+):k)+s=. ((k=. k+1) {t)+{:>{:r end. ((>0:)#])L:0 y-L:0}:r)	red=:3 :0 if. y<13 do. r=. no y else. t=. (>. k%2), 2)\$c=. i. k=. #r=. no y p=. ;(;{. "1 t) {r p; ;((s>0)#s=. {:"1 t) {r end.)
pq=:3 :0 r=. <12, 12+s=. (k=. 0) {t=. +:&. >:1+i. b=. >. y%10 while. y>{. >{:r do. r=. r, <s, (3+k)+s=. 2+((k=. k+1) {t)+{:>{:r end. qq=. r-. pp=. (+:i. >. -:#r) {r ((p<:y)#p=. ;pp); (q<:y)#q=. ;qq)	r30=:part_no"0>:i. 30 no 26 [25 24 21 19 14 11 4] 's t':=red 26 [25 24 14 11 21 19 4] pp;p=:+/pp=: (<:s) {r30 [1958 1575 135 56 3724] qq;q=:+/qq=: (<:t) {r30 [792 490 5 1287] p - q 2437] 'a b'=:pq 26 [12 15 22 26] (p-q)+(26 e. a)- 26 e. b 2436 ($n \leq 25$ までの p の値から算出) 25 { r30 2436 (確かに、n=26 のときの「p26」の値)
part_no=:3 :0 k=. #r=. 1 2 3 5 7 11 15 22 30 42 if. y<11 do. (<:y) {r else. 'p q'=. pq y while. k<y do. s=. (+/(<:>{. h) {r)-+/(<:>{:h=. red k=. k+1) {r {:r=. r, s+(k e. p)-k e. q end. end.)	

3 10 \$ part_no"0 >:i. 30	p(1) ~ p(30) の値が
---------------------------	------------------

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	示されている。
part_no"0(10*4+>:i.7) 204226 966467 4087968 15796476 56634173 190569292 607163746	分割数：P40～p100 の値

tc' p200=:x:part_no 200' 0.193761 p200 3972999029388 tc' p300=:x:part_no 300' 0.665288 p300 9253082936723528 tc' p400=:x:part_no 400' 0.554037 p400 6727090051588284416 tc' p500=:x:part_no 500' 0.803441 p500 2300165087559805829120 tc' p600=:x:part_no 600' 1.39866 p600 458005063749494810607616 tc' p700=:x:part_no 700' 1.41037 p700 60378645638998065139417088	tc' p800=:x:part_no 800' 1.76496 p800 5733259907364596643528704000 tc' p900=:x:part_no 900' 2.1649 p900 415935058092506687910289866752 tc' p1000=:x:part_no 1000' 2.61234 p1000 24070293522200282323346880724992 tc' p1500=:x:part_no 1500' 5.73519]P1500=:}. ” :p1500 892120554652152127515929766651194507264 tc' p2000=:x:part_no 2000' 9.46603]p2000=:}. ” :p2000 4242090736196891424949948886323686243393769308160 \$&> p1500;p2000 39 49
--	---

tc'part_no 20' 0.00693831	tc'part_no 100' 0.0740033	「part_no」という関数の演算時間に関しては、全くノー・プロブレムである。
tc'part_no 30' 0.0133064	tc'part_no 200' 0.182065	
tc'part_no 40' 0.0187395	tc'part_no 294' 0.332804	
tc'part_no 45' 0.0216966	tc'part_no 300' 0.358071	
x:part_no 294 6039763882095508	x:part_no 300 9253082936723528	