

### 4個のみかん：数の分割 (Partition)

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;&lt;"1(y-t),t=&gt;i.y-h=&lt;.-:ywhile.h&gt;1   do.r=.r,&lt;(s&gt;0)#s=.s,y-+/s=.(&lt;y%h)\$h=.h-1 end. )</pre>		<pre>next=:3 0 m=.:#(1=s)#s=.&gt;y h=((#s)-m){s [ t=(-m){s=.&gt;y &lt;h,+/&amp;&gt;(-&lt;:{t)&lt;\(+t)\$1 )</pre>
<pre>devide=:3 :0 t=.r=.y while.(+/\s)&gt;#(s=1)#s=.&gt;t   do.r=.r,t=.next t end. )</pre>	<pre>part=:3 :0 r=.2{s [ t=(-(#s)+&lt;:k=. _1){s=.dev y while.k&lt;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>

【スターティング関数「dev」の定義】

ヘッドの数值を斬減させながら、スタートのパターンを、ボックス内に出力する関数													
dev 9													
<table border="1"> <tr> <td>9</td><td>8 1</td><td>7 2</td><td>6 3</td><td>5 4</td><td>4 4 1</td><td>3 3 3</td><td>2 2 2 2 1</td><td>1 1 1 1 1 1 1 1 1 1</td> </tr> </table>			9	8 1	7 2	6 3	5 4	4 4 1	3 3 3	2 2 2 2 1	1 1 1 1 1 1 1 1 1 1	2 5 \$	
9	8 1	7 2	6 3	5 4	4 4 1	3 3 3	2 2 2 2 1	1 1 1 1 1 1 1 1 1 1					
dev 10													
<table border="1"> <tr> <td>10</td><td>9 1</td><td>8 2</td><td>7 3</td><td>6 4</td> </tr> <tr> <td>5 5</td><td>4 4 2</td><td>3 3 3 1</td><td>2 2 2 2 2</td><td>1 1 1 1 1 1 1 1 1 1 1 1</td> </tr> </table>				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 1 1
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 1 1									
next<5 5 <table border="1"> <tr><td>5 4 1</td></tr> </table>	5 4 1	next<5 3 2 <table border="1"> <tr><td>5 3 1 1</td></tr> </table>	5 3 1 1	next<5 2 2 1 <table border="1"> <tr><td>5 2 1 1 1</td></tr> </table>	5 2 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。							
5 4 1													
5 3 1 1													
5 2 1 1 1													
next<5 4 1 <table border="1"> <tr><td>5 3 2</td></tr> </table>	5 3 2	next<5 3 1 1 <table border="1"> <tr><td>5 2 2 1</td></tr> </table>	5 2 2 1	next<5 2 1 1 1 <table border="1"> <tr><td>5 1 1 1 1 1</td></tr> </table>	5 1 1 1 1 1								
5 3 2													
5 2 2 1													
5 1 1 1 1 1													
devide<5 5													
<table border="1"> <tr> <td>5 5</td><td>5 4 1</td><td>5 3 2</td><td>5 3 1 1</td><td>5 2 2 1</td><td>5 2 1 1 1</td><td>5 1 1 1 1 1</td> </tr> </table>			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。			
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							

「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 1
1	1

【関数「part」と「partm」の演算時間について】

「partm」が J602 の機能「M.」を利用した関数であるが、以下で示すように、そううまくは機能していない。NB. tc=:6!:2		
tc' P10=:part 10' 0.0101714 tc' P10=:partm 10' 0.000496152 # P10 42	tc' P20=:part 20' 0.110142 tc' P20=:partm 20' 0.109143 # P20 627	tc' P30=:part 30' 1.41232 tc' P30=:partm 30' 1.4044 # P30 5604
tc' P40=:part 40' 32.2255 tc' P40=:partm 40' 32.0371 # 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！

【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、メモリーがオーバーすることも無くなる。		
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	pc'n100=:partno 100' 31109 n100 190569292	

【より単純化したプログラムに変えてみる】

<pre>dev1=:3 :0 r=.&lt;"1(y-t),t=.3+i.y-2+h=.&lt;.-&gt;:y while.h&gt;2   do.r=r,&lt;(s&gt;0)#s=.s,y-+/s=.(&lt;.y%h)\$h=.h-1 end. )</pre>		<pre>dev1 7</pre> <table border="1"> <tr><td>4 3</td><td>3 3 1</td><td>2 2 2 1</td><td>dev1 8</td></tr> <tr><td>5 3</td><td>4 4</td><td>3 3 2</td><td>2 2 2 2</td></tr> </table>	4 3	3 3 1	2 2 2 1	dev1 8	5 3	4 4	3 3 2	2 2 2 2
4 3	3 3 1	2 2 2 1	dev1 8							
5 3	4 4	3 3 2	2 2 2 2							
<pre>next0=:3 :0 m=.&gt;:#(1=y)#y (h=.(#y)-m){y};t=(-m){y h,+/&amp;&gt;(-&lt;:{t)&lt;\(+t)\$1 )</pre>	<pre>devn=:3 :0 t=.y [ r=.1 while.(+}{t)&gt;#(t=1)#t   do.r=r+1 [ t=.next0 t end. )</pre>	<pre>partn=:3 :0 t=.dev1 y [ r=.6+h=._1 while.h&lt;&lt;:#t   do.r=r+devn&gt;(h=.h+1){t end. )</pre>								

partn L:0(6:7:8:9:10)  

11	15	22	30	42
----	----	----	----	----

tc'n40=:partn 40' 4.48425	tc'n50=:partn 50' 25.1907	tc'n60=:partn 60' 125.665			
n40;n50;n60 <table border="1"> <tr><td>37338</td><td>204226</td><td>966467</td></tr> </table>	37338	204226	966467	計算時間については、全く改良がみられなかった！	
37338	204226	966467			

nextm=:3 :0 M. m=.>:#(1=y)#y	devm=:3 :0 M. t=.y [ r=.1	partm=:3 :0 M. t=.dev1 y [ r=.6+h=._1
---------------------------------	------------------------------	--

<pre>(h=(#y)-m){y};t=(-m){y h,+/&amp;&gt;(-&lt;:{t}&lt;\(+/t)\$1 )</pre>	<pre>while.(+}{t}&gt;#(t=1)#t do.r=.r+1 [ t=.nextm t end. )</pre>	<pre>while.h&lt;&lt;:#t do.r=.r+devm&gt;(h=.h+1){t end. )</pre>
tc'n40=:partm 40' 4.7519	tc'n50=:partm 50' 25.1318	tc'n60=:partm 60' 129.343
「M.」を用いても全く効果なし！		

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

 $p(5) = 7$

【漸化関係が成立していない！】

$p(5) = \_1 + p(4) + p(3) = \_1 + 5 + 3 = 7$
--

$p(7) = \_1 + p(6) + p(5) - p(2) = \_1 + 11 + 7 - 2 = \_1 + 16 = 15$
$p(12) = 1 + p(11) + p(10) - \{p(7) + p(5)\} = (56 + 42) - (15 + 7) = 1 + 76 = 77$ $p(15) = 1 + \{p(14) + p(13)\} - \{p(10) + p(8)\} + p(3) = 135 + 101 - (42 + 22) + 3176$
$p(22) = \_1 + \{p(21) + p(20)\} - \{p(17) + p(15)\} + \{p(10) + p(7)\}$ $= \_1 + \{792 + 627\} - \{297 + 176\} + \{42 + 15\} = \{1419\} - \{473\} + \{57\} = 1002$ $p(26) = \_1 + \{p(25) + p(24)\} - \{p(21) + p(19)\} + \{p(14) + p(11)\} - p(4)$ $= \_1 + \{1958 + 1575\} - \{792 + 490\} + \{135 + 56\} - 5 = 3533 - 1282 + 191 - 5 = 2436$
$p(35) = 1 + \{p(34) + p(33)\} - \{p(30) + p(28)\} + \{p(23) + p(25)\} - \{p(13) + p(19)\}$ $= 1 + \{12310 + 10143\} - \{5604 + 3718\} + \{1255 + 627\} - \{101 + 30\} = 14883$ $p(40) = 1 + \{p(39) + p(38)\} - \{p(35) + p(33)\} + \{p(28) + p(25)\} - \{p(18) + p(14)\} + p(5)$ $= 1 + \{31185 + 26015\} - \{14883 + 10143\} + \{3718 + 1958\} - \{385 + 135\} + 7 = 37338$

no1 90

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

【漸化関係の成立せず、修正を余儀なくされる箇所】

6!:2'n040=:test0 40'	6!:2'n050=:test0 50'	6!:2'n060=:test0 60'
11.682	66.9439	350.662
6!:2'n40=:partno 40'	6!:2'n50=:partno 50'	6!:2'n60=:partno 60'
4.947	26.8823	137.179
n040, n40	n050, n50	n060, n60
37337 37338 (値が異なる!)	204226 204226	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】

r10	]pp=:(<:10	]qq=:(<:6 4){r10		
1 2 3 5 7 11 15 22 30 42	9){r10	11 5		
red 11	42 30	]q=:+ /qq		
<table border="1"><tr><td>10 9</td><td>6 4</td></tr></table>	10 9	6 4	]p=:+ /pp	16
10 9	6 4			
	72	p-q		
		56		

$\{r_{11}=r_{10}, p-q$ 56 's t'=:red 12 s                      t 11 10                      7 5	$]pp=:(<:s) \{r_{11}$ 56 42 ]p=:+/pp 98	$]qq=:(<:t) \{r_{11}$ 15 7 ]q=:+/qq 22	$p-q$ 76 $1+p-q$ 77
$\{r_{12}=r_{11}, 1+p-q$ 77 's t'=:red 13	$]p=:+/(<:s) \{r_{12}$ 134	$]q=:+/(<:t) \{r_{12}$ 33	$p-q$ 101
$\{r_{13}=r_{12}, p-q$ 101 's t'=:red 14	$]p=:+/(<:s) \{r_{13}$ 180	$]q=:+/(<:t) \{r_{13}$ 45	$p-q$ 135
$\{r_{14}=r_{13}, p-q$ 135 's t'=:red 15	$]p=:+/(<:s) \{r_{14}$ 239	$]q=:+/(<:t) \{r_{14}$ 64	$1+p-q$ 176
$\{r_{15}=r_{14}, 1+p-q$ 176 's t'=:red 16	$]p=:+/(<:s) \{r_{15}$ 317	$]q=:+/(<:t) \{r_{15}$ 86	$p-q$ 231
$\{r_{16}=r_{15}, p-q$ 231 's t'=:red 17	$]p=:+/(<:s) \{r_{16}$ 416	$]q=:+/(<:t) \{r_{16}$ 119	$p-q$ 297

<pre> 3 10 \$ part_no^0 &gt;:i. 30  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 part_no^0(10*4+&gt;:i. 6) 204226 966467 4087968 15796476 56634173 190569292 </pre>	
--	--

【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】

<pre> no=:3 :0  r=. &lt;1, 1+s=. (k=. 0) {t=. +:&amp;. &gt;:i. y while. y&gt;s   do. r=. r, &lt;s, (+:k)+s=. ((k=. k+1) {t})+{:r end. ((&gt;0:)#])L:0 y-L:0}:r ) </pre>	<pre> red=:3 :0  if. y&lt;13 do. r=. no y else.   t=. ((&gt;. k%2), 2)\$c=. i. k=. #r=. no y   p=. ;(;{."1 t){r   p; ;((s&gt;0)#s=. {:"1 t){r end. ) </pre>
---	---

<pre> pq=:3 :0 r=. &lt;12, 12+s=. (k=. 0) {t=. +:&amp;. &gt;:1+i. b=. &gt;. y%10 while. y&gt;{.&gt;{:r   do. r=. r, &lt;s, (3+k)+s=. 2+((k=. k+1) {t)+{:&gt;{:r end. qq=. r-. pp=. (+:i. &gt;. -:#r) {r ((p&lt;:y)#p=. ;pp);(q&lt;:y)#q=. ;qq ) </pre>	<pre> no 26 <table border="1" data-bbox="901 338 1241 376"> <tr> <td>25</td><td>24</td><td>21</td><td>19</td><td>14</td><td>11</td><td>4</td> </tr> </table> ]’ pp qq’=:red 26 <table border="1" data-bbox="901 434 1214 472"> <tr> <td>25</td><td>24</td><td>14</td><td>11</td><td>21</td><td>19</td><td>4</td> </tr> </table> r30=:part_no”0&gt;:i. 30 pp;p=:+/pp=:(&lt;:pp) {r30 <table border="1" data-bbox="901 577 1241 616"> <tr> <td>1958</td><td>1575</td><td>135</td><td>56</td><td>3724</td> </tr> </table> qq;q=:+/qq=:(&lt;:qq) {r30 <table border="1" data-bbox="901 674 1145 712"> <tr> <td>792</td><td>490</td><td>5</td><td>1287</td> </tr> </table> p - q </pre>	25	24	21	19	14	11	4	25	24	14	11	21	19	4	1958	1575	135	56	3724	792	490	5	1287
25	24	21	19	14	11	4																		
25	24	14	11	21	19	4																		
1958	1575	135	56	3724																				
792	490	5	1287																					
<pre> part_no=:3 :0 k=. #r=. 1 2 3 5 7 11 15 22 30 42 if. y&lt;11 do. (&lt;:y) {r else. ’ p q’=. pq y   while. k&lt;y     do. s=. (+/(&lt;:&gt;{.h) {r)-+/(&lt;:&gt;{:h=. red k=. k+1) {r     {:r=. r, s+(k e. p)-k e. q   end. end. ) </pre>	<pre> 2437 ]’ a b’=:pq 26 <table border="1" data-bbox="901 815 1102 853"> <tr> <td>12</td><td>15</td><td>22</td><td>26</td> </tr> </table> (p-q)+(26 e. a)- 26 e. b 2436 25 { r30 2436 </pre>	12	15	22	26																			
12	15	22	26																					

ct' p200=:x:part_no 200'	ct' p700=:x:part_no 700'
0. 193761	1. 41037
p200	p700
3972999029388	60378645638998065139417088
ct' p300=:x:part_no 300'	ct' p800=:x:part_no 800'
0. 665288	1. 76496
p300	p800
9253082936723528	5733259907364596643528704000
ct' p400=:x:part_no 400'	ct' p900=:x:part_no 900'
0. 554037	2. 1649
p400	p900
6727090051588284416	415935058092506687910289866752
ct' p500=:x:part_no 500'	ct' p1000=:x:part_no 1000'
0. 803441	2. 61234
p500	p1000
2300165087559805829120	24070293522200282323346880724992
ct' p600=:x:part_no 600'	ct' x:p2000=:part_no 2000'
1. 39866	9. 83628
p600	x:p2000
458005063749494810607616	4242090736196891424949948886323686243393769308160

メデタシ メデタシ!