Parcels

Gustavo Delfino, 2013-07-08

Out[7]=

Definitions:

Show examples

1	noun	a wrapped container
2	noun	the allotment of some amount by dividing something
3	noun	an extended area of land
4	noun	a collection of things wrapped or boxed together
5	verb	divide into parts
6	verb	cover with strips of canvas
7	verb	make into a wrapped container

(7 meanings)

The Excercise

This would be the exercise:

1. Find the most efficient combination of 3kg and 5kg parcels required to make any shipment of 8kg or more. The app will accept input from a web form.
2. Output should either be a simple result view or for bonus points, rendered on the same page via AJAX without page refresh. So given 11, output should be:
   - 1 five kg parcels
   - 2 three kg parcels
3. Input should be validated.
4. Assume at least Ruby 1.9.3 and Rails 3+. Please use TDD and submit your answer with either a private github repo or via email.

Manual Procedure

Automating the Search of a Solution

Function to calculate the parsels

In[9]:=
```
parsels[n_] := NestWhile[{n - 5 (⌊n/5⌋ - #[[3]]), 5 (⌊n/5⌋ - #[[3]]), #[[3]] + 1} &,
    {1, n, 0}, (! Divisible[#[[1]], 3]) &] /. {iii_, v_, _} :> {v / 5, iii / 3}
```

```
parsels[99]
```

{18, 3}

Function to pretty print the parsels

Table of Results

```
Table[{n, printParsels[parsels[n]]}, {n, 8, 100}] // TableForm
```

8	5×1 + 3×1
9	3×3
10	5×2
11	5×1 + 3×2
12	3×4
13	5×2 + 3×1
14	5×1 + 3×3
15	5×3
16	5×2 + 3×2
17	5×1 + 3×4
18	5×3 + 3×1
19	5×2 + 3×3
20	5×4
21	5×3 + 3×2
22	5×2 + 3×4
23	5×4 + 3×1
24	5×3 + 3×3
25	5×5
26	5×4 + 3×2
27	5×3 + 3×4
28	5×5 + 3×1
29	5×4 + 3×3
30	5×6
31	5×5 + 3×2
32	5×4 + 3×4
33	5×6 + 3×1
34	5×5 + 3×3
35	5×7
36	5×6 + 3×2

37	5×5 + 3×4
38	5×7 + 3×1
39	5×6 + 3×3
40	5×8
41	5×7 + 3×2
42	5×6 + 3×4
43	5×8 + 3×1
44	5×7 + 3×3
45	5×9
46	5×8 + 3×2
47	5×7 + 3×4
48	5×9 + 3×1
49	5×8 + 3×3
50	5×10
51	5×9 + 3×2
52	5×8 + 3×4
53	5×10 + 3×1
54	5×9 + 3×3
55	5×11
56	5×10 + 3×2
57	5×9 + 3×4
58	5×11 + 3×1
59	5×10 + 3×3
60	5×12
61	5×11 + 3×2
62	5×10 + 3×4
63	5×12 + 3×1
64	5×11 + 3×3
65	5×13
66	5×12 + 3×2
67	5×11 + 3×4
68	5×13 + 3×1
69	5×12 + 3×3
70	5×14
71	5×13 + 3×2
72	5×12 + 3×4
73	5×14 + 3×1
74	5×13 + 3×3
75	5×15
76	5×14 + 3×2
77	5×13 + 3×4
78	5×15 + 3×1
79	5×14 + 3×3
80	5×16
81	5×15 + 3×2
82	5×14 + 3×4
83	5×16 + 3×1
84	5×15 + 3×3
85	5×17
86	5×16 + 3×2
87	5×15 + 3×4
88	5×17 + 3×1
89	5×16 + 3×3
90	5×18
91	5×17 + 3×2

92	5×16 + 3×4
93	5×18 + 3×1
94	5×17 + 3×3
95	5×19
96	5×18 + 3×2
97	5×17 + 3×4
98	5×19 + 3×1
99	5×18 + 3×3
100	5×20

10000 Tests !

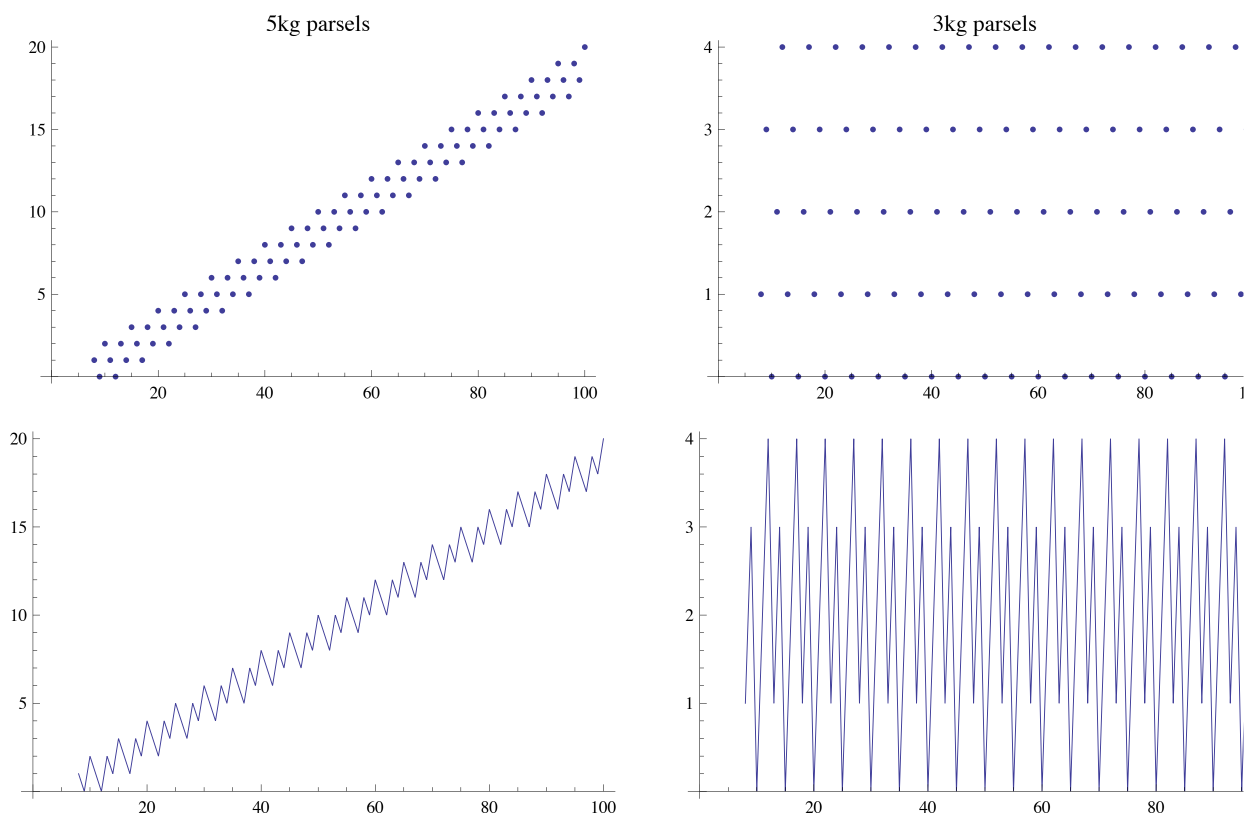
Test from 8kg to 10000kg

```
And @@ Table[n == 5 parsels[n]⟦1⟧ + 3 parsels[n]⟦2⟧, {n, 8, 10 000}]
```

```
True
```

Plots

```
GraphicsGrid[
 {{ListPlot[{Table[{n, parsels[n][[1]]}, {n, 8, 100}]}, PlotLabel → "5kg parsels"],
   ListPlot[
    {Table[{n, parsels[n][[2]]}, {n, 8, 100}]}, PlotLabel → "3kg parsels"]},
  {ListPlot[{Table[{n, parsels[n][[1]]}, {n, 8, 100}]}, Joined → True],
   ListPlot[{Table[{n, parsels[n][[2]]}, {n, 8, 100}]}, Joined → True]}}]
```



Iteration Free Algorithm

```
def kg= val
  # This algorithm was deduced with the help of Wolfram Mathematica
  self.units3kg = (2 * val - 10) % 5
  self.units5kg = ([-3,-9,0,-6,-12].rotate(val-8)[0] + val)/5
  super val
end
```

3kg Units

These are the first few terms of the 3kg sequence:

```
Table[parsels[n]⟦2⟧, {n, 8, 20}]
```

```
{1, 3, 0, 2, 4, 1, 3, 0, 2, 4, 1, 3, 0}
```

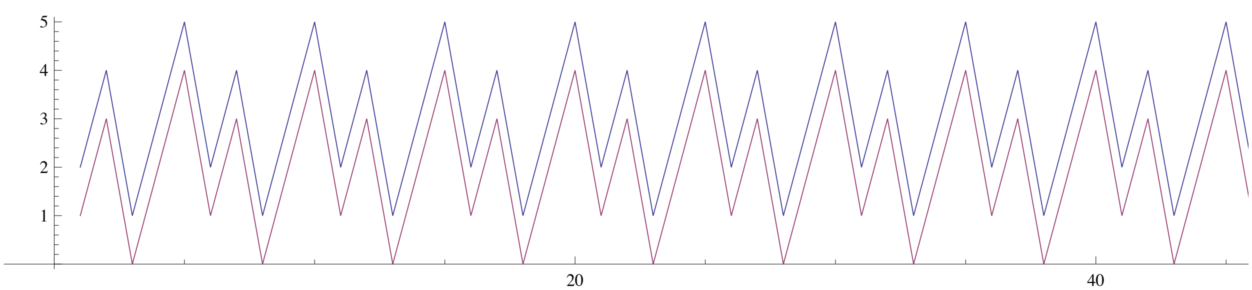
Mathematica finds the generating function directly!

```
FindSequenceFunction[%]
```

```
Mod[4 + 2 #1, 5] &
```

As a verification we will plot this function over the original function with an offset of 1:

```
ListPlot[
 {Table[parsels[n][[2]] + 1, {n, 8, 100}],
  Table[Mod[2 # - 10, 5] &[n], {n, 8, 100}]}, Joined → True, AspectRatio → 1 / 10]
```



5kg Units

These are the first few terms of the 5kg sequence:

```
Table[parsels[n]⟦1⟧, {n, 8, 20}]
```

```
{1, 0, 2, 1, 0, 2, 1, 3, 2, 1, 3, 2, 4}
```

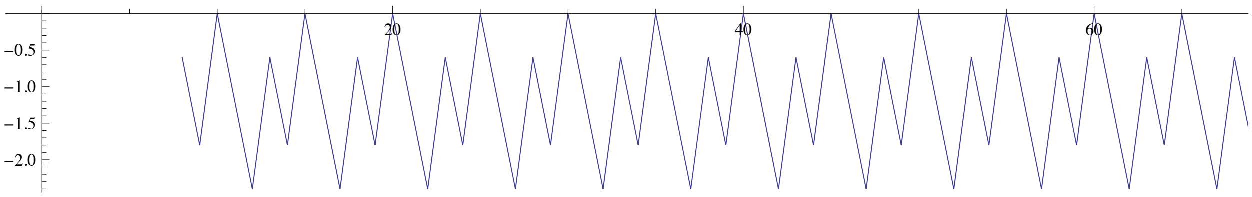
This time we are not so lucky as *Mathematica* can’t find a simple closed for solution for our sequence. But we can with a little work.

```
FindSequenceFunction[%]
```

```
DifferenceRoot[Function[{y, n},
  {-1 - y[n] + y[5 + n] == 0, y[1] == 1, y[2] == 0, y[3] == 2, y[4] == 1, y[5] == 0}]]
```

Lets remove the slope from this sequence by substracting $n/5$ from it:

```
ListPlot[Table[{n, parsels[n][[1]] - n / 5}, {n, 8, 100}],
 Joined → True, AspectRatio → 1 / 10]
```




```
Table[parsels[n][[1]] - n / 5, {n, 8, 50}]
```

$$\left\{-\frac{3}{5}, -\frac{9}{5}, 0, -\frac{6}{5}, -\frac{12}{5}, -\frac{3}{5}, -\frac{9}{5}, 0, -\frac{6}{5}, -\frac{12}{5}, -\frac{3}{5}, -\frac{9}{5}, 0, -\frac{6}{5}, -\frac{12}{5}, -\frac{3}{5}, -\frac{9}{5}, 0, -\frac{6}{5}, -\frac{12}{5}, -\frac{3}{5}, -\frac{9}{5}, 0, -\frac{6}{5}, -\frac{12}{5}, -\frac{3}{5}, -\frac{9}{5}, 0, -\frac{6}{5}, -\frac{12}{5}, -\frac{3}{5}, -\frac{9}{5}, 0, -\frac{6}{5}, -\frac{12}{5}, -\frac{3}{5}, -\frac{9}{5}, 0, -\frac{6}{5}, -\frac{12}{5}, -\frac{3}{5}, -\frac{9}{5}, 0\right\}$$

Mathematica now can find the generating function, but it is too complicated...

```
FindSequenceFunction[%] // Simplify
```

$$-\frac{6}{5} - \frac{12}{25}\,\mathrm{Cos}\left[\frac{2}{5}\pi\,(-5+\#1)\right] - \frac{12}{25}\,\mathrm{Cos}\left[\frac{4}{5}\pi\,(-5+\#1)\right] - \frac{12}{25}\,\mathrm{Cos}\left[\frac{6}{5}\pi\,(-5+\#1)\right] - \frac{12}{25}\,\mathrm{Cos}\left[\frac{8}{5}\pi\,(-5+\#1)\right] - \frac{6}{25}\,\mathrm{Cos}\left[\frac{2}{5}\pi\,(-4+\#1)\right] - \frac{6}{25}\,\mathrm{Cos}\left[\frac{4}{5}\pi\,(-4+\#1)\right] - \frac{6}{25}\,\mathrm{Cos}\left[\frac{6}{5}\pi\,(-4+\#1)\right] - \frac{6}{25}\,\mathrm{Cos}\left[\frac{8}{5}\pi\,(-4+\#1)\right] - \frac{9}{25}\,\mathrm{Cos}\left[\frac{2}{5}\pi\,(-2+\#1)\right] - \frac{9}{25}\,\mathrm{Cos}\left[\frac{4}{5}\pi\,(-2+\#1)\right] - \frac{9}{25}\,\mathrm{Cos}\left[\frac{6}{5}\pi\,(-2+\#1)\right] - \frac{9}{25}\,\mathrm{Cos}\left[\frac{8}{5}\pi\,(-2+\#1)\right] - \frac{3}{25}\,\mathrm{Cos}\left[\frac{2}{5}\pi\,(-1+\#1)\right] - \frac{3}{25}\,\mathrm{Cos}\left[\frac{4}{5}\pi\,(-1+\#1)\right] - \frac{3}{25}\,\mathrm{Cos}\left[\frac{6}{5}\pi\,(-1+\#1)\right] - \frac{3}{25}\,\mathrm{Cos}\left[\frac{8}{5}\pi\,(-1+\#1)\right]\ \&$$

The linearized sequence is greatly simplified by multiplicating it by 5:

```
5 Table[parsels[n][[1]] - n / 5, {n, 8, 50}]
```

```
{-3, -9, 0, -6, -12, -3, -9, 0, -6, -12, -3, -9, 0, -6, -12, -3, -9, 0, -6, -12, -3, -9,
 0, -6, -12, -3, -9, 0, -6, -12, -3, -9, 0, -6, -12, -3, -9, 0, -6, -12, -3, -9, 0}
```

And this is just a repeating pattern:

```
Table[First[RotateLeft[{-3, -9, 0, -6, -12}, n - 8]], {n, 8, 50}]
```

```
{-3, -9, 0, -6, -12, -3, -9, 0, -6, -12, -3, -9, 0, -6, -12, -3, -9, 0, -6, -12, -3, -9,
 0, -6, -12, -3, -9, 0, -6, -12, -3, -9, 0, -6, -12, -3, -9, 0, -6, -12, -3, -9, 0}
```

Reversing the transformations:

$$\text{Table}\left[\frac{\text{First}[\text{RotateLeft}[\{-3, -9, 0, -6, -12\}, n - 8]]}{5}, \{n, 8, 50\}\right]$$

$$\left\{-\frac{3}{5}, -\frac{9}{5}, 0, -\frac{6}{5}, -\frac{12}{5}, -\frac{3}{5}, -\frac{9}{5}, 0, -\frac{6}{5}, -\frac{12}{5}, -\frac{3}{5}, -\frac{9}{5}, 0, -\frac{6}{5}, -\frac{12}{5}, -\frac{3}{5}, -\frac{9}{5}, 0, -\frac{6}{5}, -\frac{12}{5}, -\frac{3}{5}, -\frac{9}{5}, 0, -\frac{6}{5}, -\frac{12}{5}, -\frac{3}{5}, -\frac{9}{5}, 0, -\frac{6}{5}, -\frac{12}{5}, -\frac{3}{5}, -\frac{9}{5}, 0, -\frac{6}{5}, -\frac{12}{5}, -\frac{3}{5}, -\frac{9}{5}, 0, -\frac{6}{5}, -\frac{12}{5}, -\frac{3}{5}, -\frac{9}{5}, 0\right\}$$

$$\text{Table}\left[\frac{\text{First}[\text{RotateLeft}[\{-3, -9, 0, -6, -12\}, n - 8]] + n}{5}, \{n, 8, 50\}\right]$$

```
{1, 0, 2, 1, 0, 2, 1, 3, 2, 1, 3, 2, 4, 3, 2, 4, 3, 5, 4, 3,
 5, 4, 6, 5, 4, 6, 5, 7, 6, 5, 7, 6, 8, 7, 6, 8, 7, 9, 8, 7, 9, 8, 10}
```

And this matches the original iterative function:

```
Table[parsels[n][[1]], {n, 8, 50}]
```

```
{1, 0, 2, 1, 0, 2, 1, 3, 2, 1, 3, 2, 4, 3, 2, 4, 3, 5, 4, 3,
 5, 4, 6, 5, 4, 6, 5, 7, 6, 5, 7, 6, 8, 7, 6, 8, 7, 9, 8, 7, 9, 8, 10}
```

Data for Ruby Test

In order to genetare data for our tests, just run this command:

```
CopyToClipboard[ExportString[Table[Join[{kg}, parsels[kg]], {kg, 8, 100, 1}], "CSV"]]
```

and the paste into your text editor.

Decor

In[31]:=
```
g1 = Rotate[Column[Grid[{#}, Spacings → 0] & /@ Table[parsels[n], {n, 8, 50}] /.
      {v_, iii_} :→ Flatten[{Table[3, {iii}], Table[5, {v}]}] /.
     {5 → Graphics[{FaceForm[LightGray], EdgeForm[{LightGray, AbsoluteThickness[
            5]}], Rectangle[{0, 0}, {5, 3}]}, ImageSize → {50, 30}],
      3 → Graphics[{FaceForm[White], EdgeForm[{LightGray, AbsoluteThickness[5]}],
         Rectangle[{0, 0}, {3, 3}]}, ImageSize → {30, 30}]},
   Alignment → Left, Spacings → 0], π / 2]
```

Out[31]=

In[32]:=
```
Export["~/Sites/parcels/app/assets/images/bg.png", g1, Background → None]
```

Out[32]= ~/Sites/parcels/app/assets/images/bg.png