Generation Lock – Analysis of Number of Moves to Solve

by Dr. Goetz Schwandtner 30.06.2013
http://puzzles.schwandtner.info
goetz@schwandtner.info

0. Introduction

The Generation Lock is a bigger version of the "Lock 250+", both by Jean-Claude Constantin. Both have rivets that interact with sliders in the innards of the lock and to open the lock, all sliders have to be moved from start to finish (right to left for Generation Lock in the picture).

Now some mathematical analysis before we provide the result – skip if you don't like this

1. Puzzle structure

The Generation Lock is 15-ary:

- There are 8 rivets/sliders 0,...,7 from bottom to top
- Each rivet/slider (but the lowest) have 15 positions 0,...,14 from right to left
- Uniform condition for moves:
  - For a rivet/slider i to move between positions j and j+1
    - if j is odd, all lower rivets/sliders have to be in position 14
    - if j is even, all lower rivets/sliders have to be in position 0

This condition already offers us a recursive algorithm to solve the lock and to estimate the number of moves, which I will present in some pseudo code:

Solve n-slider lock:

```
1. if n=1:
2.      slide slider 0 to position 1 (leftmost, as slider 0 has only position 0 and 1)
3. else:
4.      For i = 0 to 13:
5.           if i is even:
```

```
6.                  move all sliders below (0,...,n-1) to position 14
7.            else: // i is odd then
8.                  move all sliders below (0,...,n-1) to position 0
9.            move slider n from i to i+1
10.     move all sliders below (0,...,n-1) to position 14
```

Some comments:
After we have moved slider n to position 14 in the last loop iteration (i=13), the loop variable i was odd, hence all the sliders below are in position 0. We have to move them to the left (position 14) afterwards, as all should be in position 14 after the algorithm has finished. This is done in line 10.

For this algorithm, we have not yet explained how to implement the very similar sub-algorithms in lines 6, 8, and 10. In fact, they are just recursive calls:

Line 6, 10:
“move all sliders below (0,...,n-1) to position 14” is done by calling our recursive algorithm **Solve n-slider lock** for value n-1: Solve n-1 slider lock

Line 8:
Our solving algorithm moves all sliders from 0 to 14. So if we run it backwards, then it will move all sliders from 14 to 0. So for “move all sliders below (0,...,n-1) to position 0” we use the same algorithm as above, but the loop reversed, the move in line 9 reversed, and of course 0 and 14 interchanged in lines 6, 8, and 10.

This algorithm will solve our lock. I leave proof of correctness and termination to the reader.

2. Solution Length – calculating the formula

We denote the running time of the algorithm above by T(n). So T is a function that gives us the exact number of moves for the solution, depending on the number n of sliders.

The algorithm above tells us the recursion scheme to derive T(n):

Special case for one slider, n=1:
$T(1)=1$
Case for n>1:
$T(n)= 15 * T(n-1) + 14$

To see this, look at the number of moves in the algorithm lines:

Lines	Moves	Comments
1 – 2	1	Special case T(1)=1
6	7*T(n-1)	Recursive call for n-1 sliders, in 7 even of the 14 loop iterations
8	7*T(n-1)	Recursive call for n-1 sliders, in 7 odd of the 14 loop iterations
9	14	One move per loop iteration

4-9	14*T(n-1)+14	Loop aggregated (over lines 4 to 9)
10	T(n-1)	Recursive call for n-1 sliders
3-10	15*T(n-1)+14	Sum over all lines in the recursive part

Now do some little exercise like back in university times to solve this recursion:

$$\begin{aligned}
T(n) &= 15\ T(n-1) + 14 \\
&= 15\ (15\ T(n-2) + 14) + 14 && \text{// substituting } T(n-1)=15T(n-2)+14 \\
&= 15^2\ T(n-2) + 15*14 + 14 \\
&= 15^2\ (15\ T(n-3) + 14) + 15*14 + 14 && \text{// substituting } T(n-2)=15T(n-3)+14 \\
&= \ldots. \\
&= 15^k\ T(n-k) + \Sigma_{j=0,\ldots,k-1}\ 15^j *14 \\
&= \ldots. \\
&= 15^{n-1}\ T(1) + \Sigma_{j=0,\ldots,n-2}\ 15^j *14 && \text{// when k reaches n-1} \\
&= 15^{n-1} + 14*\ \Sigma_{j=0,\ldots,n-2}\ 15^j && \text{// note T(1)=1 as above} \\
&= 15^{n-1} + 14*\ ((15^{n-1} -1)/(15-1)) && \text{// well known explicit formula for geometric series} \\
&= 15^{n-1} + 14*\ ((15^{n-1} -1)/14) \\
&= 15^{n-1} + (15^{n-1} -1) \\
&= 15^{n-1} + 15^{n-1} -1 \\
&= 2*15^{n-1} -1
\end{aligned}$$

So this is the explicit formula:

$$T(n)=2*15^{n-1} -1$$

3. Solution Length – result

What we have neglected so far, is that we will have to pull out the shackle at the end, so the total number of moves will be one higher:

Total number of moves to solve lock with n rivets $2*15^{n-1}$

In our case, we have n=8, so the explicit number of moves will be:

$2*15^{8-1} =2*15^7 = 341\ 718\ 750$

In other words: ***over 340 million moves.***

4. Conclusion

This will take a while.[1]

Happy puzzling!

---

1 Assuming we employ a group of puzzlers willing to work full time on this lock, without holidays and in shifts, so that the solving is going on 24 hours a day, and 365 days a year, and on average 2 seconds per move, this will take about 21.6 years. (now corrected from first version)