**COSC241 Assignment - Deal with It Report**

This report answers the questions outlined in the ‘Deal with It’ assignment handout regarding possible experiments and investigations made through the code.

***The questions and answers are as followed:***

1. **Consider the count values resulting from the “pick up by rows” specifications (those beginning with an L or R). What values do they take and why?**

*For this question we refer to each row as “sets” as it gives the best visual description to describe picking up the cards in order to transform them.*

Given the specifications of LT, LB, RT or RB, they will always take the values either 1 or 2. This is because it will always pick up the cards in a “set” of row amount n. For example, if ***we have size = 6, rowLength = 2, spec = RB***; we will always be picking up in “sets” of 3, meaning from 6-4 and then 3-1(A), then transforming them (placing them back down) as 1(A)-3 then 4-6. After this it simply takes a transformation of ***1 or 2 times*** to get back to the original 1 to 6 (1 to *n*) placement of cards.

Overall, if it is from the left, the contents of the “set” will be in the correct order while transforming. If it’s from the right, it’ll be in the reverse. Just the same if it’s from the top, the order will be correct within the “set”, while when from the bottom, it’ll be the reverse. So, it’s not necessarily the contents of the “set” that will change; by starting from L or R, it’s not ever fully “shuffled”. Therefore, since you are only changing to reverse order for the contents of “set” transformed in RT, or reverse order of “sets” in LB or both, the contents and the order of the “sets” will change in the case of RB. In the worst-case scenario, it will take at most two iteration to get back to the original pile state (array) since you’re simply reversing the order, and the second transformation you are just reversing back to what you started with.

Best case scenario, it won’t change the order, for example: size 6, rowLength 1. If we use RT it won’t reverse the order at all, as it is literally a set of 1. If you reverse the set it won’t change the order if it is one number

1. **What is the maximum count value produced for any specification and any pile size of 20 or less? What pile size(s), row length(s) and specification(s) produce it? Given a pile size, row length, and specification can you think of a way of computing its count that doesn’t rely on actually carrying out that many transformations?**

The maximum count value produced for any specification and pile of 20 or less is 18. The variable specifications that produce a maximum count of 18 are:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CP 20 2** | **CP 20 4** | **CP 20 5** | **CP 20 10** | **CP 18 2** | **CP 18 3** | **CP 18 6** | **CP 18 9** |
| TL 18 | BR 18 | BR 18 | TL 18 | TR 18 | TR 18 | BL 18 | BL 18 |
| BR 18 |  |  | BR 18 |  |  |  |  |

A way to compute this count without carrying out that many transformations is

1. **There are 720 possible card piles consisting of the numbers 1 through 6 in some order. Call such a pile accessible if it can be reached from the original pile 123456 by some sequence of transformations. How many accessible piles are there? What about seven, eight or nine card piles? For how large a value of *n* do you think it might be feasible to compute the number of accessible piles (and why)?**