Alex Lake-Smith 5400306
Connor Spear 8262050

Question 1:

Specifications that pick up by rows, only take 2 or fewer counts to revert back to their original pile state no matter the pile or row length. This is because the first transformation rearranges the pile of cards to the chosen transformation specification which is always a flipped copy of the pile starting at the specifications start point, the second transformation then flips the pile back to the original state as the start point is now card number 1. We can see this pattern occur across all pile and row lengths no matter how long they are. Picking up by rows will always flip the card pile at the starting point in a particular way depending on its specification, therefore the second count will flip the transformed pile state to the original pile state at the starting point.

Transformation spec: RT				Transformation spec: RB		
1	2	3		1	2	3
4	5	6		4	5	6
7	8	9		7	8	9
3	2	1		9	8	7
6	5	4		6	5	4
9	8	7		3	2	1

We can see that the specification 'RT' has flipped horizontally across the values 2, 5 and 8 whereas the specification 'RB has flipped around the value 5, although each pick up by row specification changes differently, they all retain a count of 2 as they flip "there and back". This can be represented as the 'big o notation' $O(1)$, as these particular transformations will always execute in the same time and space without regard for size of piles or row lengths.

Question 2:

The maximum count value produced for any specification and pile size of up to 20 cards is 18 transformations to revert back to its original pile state. As it is specified that the row length must be a multiple of the pile size, that immediately tells us our highest count values will be returned from even piles. As we've already found, picking up by row only specifications only require 2 transformations to return its original state, we expected that the picking up by column specifications would return a higher count.

There were multiple different specifications and pile sizes that produced 18 counts. For card piles of size 20, the BR specification returned 18 counts for all row lengths, TL also returned 18 counts for row lengths 10 and 2. For card piles sizes of 18, the TR specification returned 18 counts with a row length of 3, whereas the BL specification returned the same 18 counts but with a row length of 6. No cards with a pile size below 18 produced a count as high.

Question 3:

To compute the piles in order of 1 through n in some order we must take the factorial of n, this can be represented in ‘big o notation’ as $O(n!)$, this number can reach extremely high values, extremely quick. Once the pile size gets to 20 there is a possible 2,432,902,008,176,640,000 different combinations that are accessible through various orders. Today most computers are run on 64-bit processors which can only store up to 18,446,744,073,709,551,616 or 2^{64} values. This would mean computing accessible piles over 20 would overflow a long value and therefore any values output after would be incorrect. Even though a computer could handle computing piles up to 20 the time it would take to compute all these possible combinations would be significantly slower than piles with a smaller size.