BRIEF INTRODUCTION ON STACK SORTING

LIM, CONG HAN

1. Introduction to the Stack-Sorting procedure

The problem of stack sorting was introduced by Knuth in the 1960s. He described the stack sorting operation as the movement of railway cars across a railroad switching network. The problem is also similar to the childhood game of Hanoi Towers, in which the player is supposed to move concentric discs of varying sizes from one side across to another without placing a bigger disc above a smaller one. This paper aims to introduce the basic concept of stack sorting to the reader.

A detailed description of the operation is as follows:
Consider the n-sized permutation $\pi = a_1 a_2 \dots a_{n-1} a_n$. This permutation is known as the 'input'. The only tool we have for sorting is a 'stack', a vertical array. In the first step, we place a_1 into the stack. For the second step, we now compare it with the element a_2. If $a_1 > a_2$ then we place a_2 on the stack above a_1, otherwise, we shall place a_1 into the output and place a_2 on top of the stack.

Subsequently, for the each step, we compare the left-most element in the input with the element on the top of the stack. The process ends when all the elements have been placed into the output stack.

We shall illustrate this with an example.

Example 1.1. Consider the permutation $\pi = 2413$

$\therefore$ Refer to Table 1.

If the image $s(\pi)$ is the identity permutation (i.e. $s(\pi) = a_{1'} a_{2'} \dots a_{n'}$ such that $a_{1'} < a_{2'} < \dots < a_{n'}$), then we say that the permutation π is *one stack-sortable*.

We now introduce the recursive definition of the stack sorting operation:

Theorem 1.2. *Consider the permutation $\pi = a_1 a_2 \dots a_{n-1} a_n$. Let $x = max\{a_1, a_2, \dots, a_{n-1}, a_n\}$. Let π_L and π_R be the terms such that $\pi = \pi_L x \pi_R$. Then*

$$s(\pi) = s(\pi_L) s(\pi_R) x$$

Table 1. Example of Stack Sorting

Step	Input	Stack	Output
Initial	2413		
1	413	2	
2	413		2
3	13	4	2
4	3	14	2
5	3	4	21
6		34	21
7		4	213
8			2134

Proof.
This is trivial. Every element before x will enter and leave the stack (and hence π_L will be sorted) before x enters since it is larger. Likewise, after x enters the stack, every element after x will enter and leave the stack before x can leave. Hence our claim holds.

□

2. One Stack Sortable Permutations

We shall now introduce a notation for describing certain patterns contained within the permutation:
If the elements a, b and c occur in the permutation π where $a < b < c$ and b precedes c precedes a, then we say that π contains a 231-pattern.

Theorem 2.1. *A permutation π is one stack sortable if and only if it does not contain a* 231*-pattern.*

Proof. If the permutation π contains a 231-pattern, then under the recursive definition of stack sorting, since $s(\pi_L)$ will contain an element that is larger than some element in $s(\pi_R)$, hence the image is not an indentity permutation.

Consider the case if a permutation does not contain tbe 231-pattern.
For any 2 elements a and b such that a precedes b, if $a > b$ then $\nexists c$ such that c is between a and b and $c > a$ (avoiding the 231-pattern). Thus, a will enter the stack and not leave till b has left the stack, hence b will precede a in the image $s(\pi)$. If $a < b$ then a will enter and leave the stack before b enters, hence, hence a will precede b in the image. Hence, the image will be the identity pattern, so π is one stack sortable.

□

Knuth proved that the number of n-permutations which are one stack sortable is the Catalan number C_n by considering the reverse operation starting from an identity permutation. Here we will prove it directly.

Theorem 2.2. *The number of one stack sortable n-permutations is the Catalan number* C_n

Proof. We know that every permutation which avoids a 231-pattern is sortable. We define $f(n)$ to be the number of one stack sortable n-permutations and $f(0) = 1$. Consider the n-permutation $\pi_n = a_1 a_2 \dots a_{n-1} a_n$ and the element $x = \max\{a_1 a_2 \dots a_{n-1} a_n\}$ such that $\pi_n = \pi_L x \pi_R$. By theorem 2.1, every element on the left of x must be smaller than every element on the right of x. Hence, the number of sortable permutations must be the number of sortable sub-permutations on the left of x multiplied by the number of sortable sub-permutations on the right of x. Summing all the possible positions of the largest element x, we get:

$$f(n) = \sum_{i=1}^{n} f(i-1) f(n-i)$$

This is analogous to the recurrence relation that generates the catalan number:

$$C_0 = 1 \text{ and } C_n = \sum_{i=1}^{n} C_{i-1} C_{n-i}$$

□

3. Other observations of Stack Sortable Permutations

Because of the 231-pattern limitation, many permutations are not one stack sortable. To increase the number of sortable permutations, we can take the image and sort it again with the stack. If this new image is the identity permutation, then we say that the stack is two stack sortable.

We shall lay down some of the properties of two stack sortable permutations:

Theorem 3.1. *A permutation* π *is two stack sortable if and only if it does not contain a* 2341*-pattern and does not contain a* 3241*-pattern which of not part of a* 35241*-pattern.*

Proof. First we want to show that π is not two stack sortable if it fulfills the above conditions.

Assume the elements $a, b, c, d \in \pi$ where $a < b < c < d$ form a 2341-pattern. Elements c, d and a form a 231-pattern, hence after one stack sorting, c will still precede a in the image. Furthermore, because b precedes c in π and $b < c$ then b will still precede c in the image $s(\pi)$ (thm 2.1). Hence, a, b, c form a 231-permutation in $s(\pi)$.
Now consider the case where the elements $w, x, y, z \in \pi$ where $w < x < y < z$

form a 3241-pattern which is not part of a 35241-pattern. There are 2 cases:
Case One: If there are no entries between x and y that are larger than both (i.e. not 35241 or 34251-pattern) then by the same logic as theorem 2.1, x will precede y in $s(\pi)$ and hence a 231-pattern is formed in $s(\pi)$.
Case Two: If there is an entry m between x and y such that $x < y < m < z$ (i.e. 34251-pattern) then y, t, z, w will form a 2341-pattern in $s\pi$).

Now we need to show that when if π is not two stack sortable then it will contain at least one of the above patterns.

If π is not two stack sortable then $s(\pi)$ contains a 231-pattern formed by elements $e, f, g \in \pi$ where $e < f < g$. By the logic in theorem 2.1, e must occur after f and g in π and there must be some element $h > g$ such that h seperates g and f from e. If f preceded g in π then π contains a 2341-pattern. If g precedes f in π then since f precedes g in $s(\pi)$ there is no entry between f and g in π that is greater than both. Hence, π contains a 3241-pattern that is not part of a 35241-pattern.

□

The number of 2 stack sortable n-permutations was conjectured by West to be $\frac{2(3n)!}{(n+1)!(2n+1)!}$. This conjecture was first proven by D. Zeilberger and subsequently there have been a few other proofs invovling the use of bijections. For this paper a proof will not be shown.

Corollary 3.2. *If the permutation π contains $q_k = 234\ldots k1$ as a pattern, then π is not (k-2) stack sortable.*

Proof. Prove by induction on k. We proved in thm 2.1 that the statement holds for the base case where $k = 3$. For the induction step, $s(\pi)$ will contain q_{k-1}.

□

4. Further Reading

This paper was intended to be a brief introduction to stack sorting. The interested reader should refer to "A Survey of Stack Sorting Disciplines" by Miklos Bona for further reading.

References

[1] Miklos Bona. A Survey Of Stack Sorting Disciplines. Electronic Journal of Combinatorics, 9 (2), 2002-2003.
[2] D. E. Knuth. "The Art of Computer Programming", volume 1, Fundamental Algoriths. Addison-Wesley, (1973).
[3] D. Zeilberger. A proof of Julian West's conjecture that the number of two-stack-sortable permutations of length n is $2(3n)!/((n+1)!(2n+1)!)$ Discrete Math., 102 (1992), 85-93.