Sorting with a Forklift

M. H. Albert and M. D. Atkinson

Department of Computer Science, University of Otago

What is a fork stack?

- Why can't we lift more than one dish from a stack?
- Why can't we put more than one dish onto a stack?

What is a fork stack?

- Why can't we lift more than one dish from a stack?
- Why can't we put more than one dish onto a stack?

- A *fork stack* is a stack whose push and pop operations are allowed to accept, or produce, sequences of elements.

What is a fork stack?

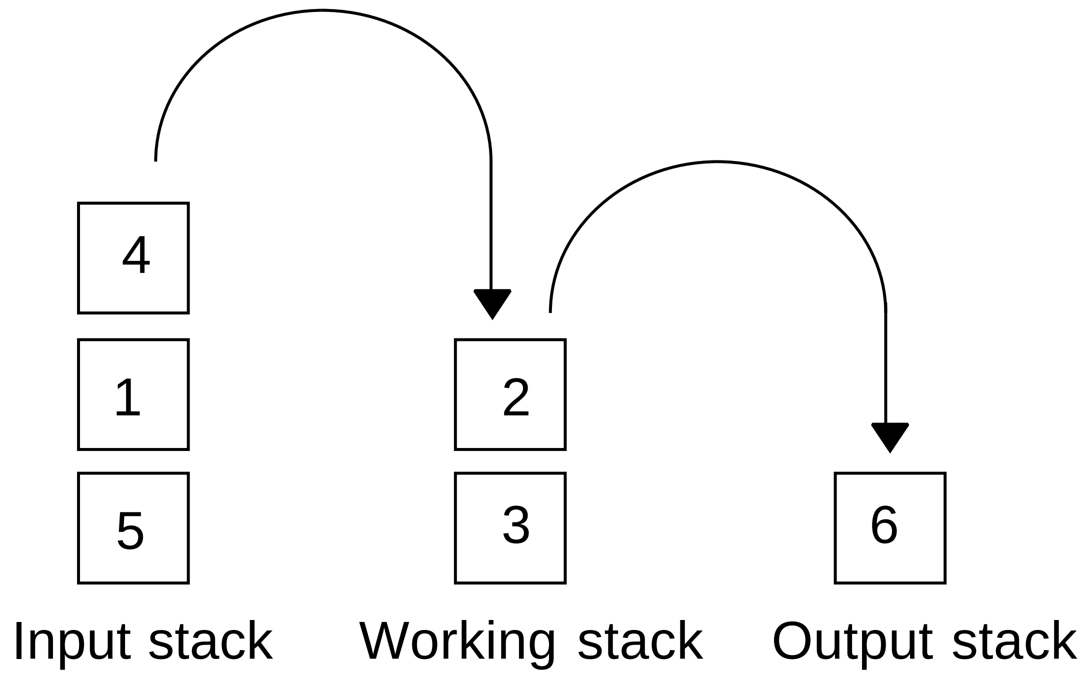
- Why can't we lift more than one dish from a stack?
- Why can't we put more than one dish onto a stack?

- A *fork stack* is a stack whose push and pop operations are allowed to accept, or produce, sequences of elements.

- A useful generalisation for example in merging the contents of two stacks.

Sorting with a single forkstack

A snapshot of sorting in progress using a single forkstack (the working stack).



To complete the sort, move the pair 41 to the working stack, move 5 to the working stack and then to output, move 4 to output, and move the triple 123 to output.

When does sorting fail?

Definition: *For positive integers a and b, $a << b$ means $a < b - 1$. In a series of fork stack moves, we say that the dreaded 13 occurs if at some point the working stack contains adjacent elements ab with $a << b$. A sequence is near-decreasing if it is decreasing except possibly for some steps of $+1$.*

Proposition: *A permutation π is unsortable if and only if every allowable sequence of fork stack operations that empties the input produces, at some point, the dreaded 13.*

An algorithm for sorting

repeat

Perform as much output as possible.

Move the maximal near-decreasing sequence from the input stack to the working stack.

until *input stack is empty*

if *working stack is empty* **then**

Success!

else

Failure.

end if

Obstructions to sorting

- There is a finite set of patterns which, if any one of them occurs in the input stack, prevent successful sorting.
- Conversely, if none of them occur in the input stack then sorting will succeed.
- The minimal such set consists of one pattern of length 5 (35142), 45 patterns of length six, and 6 of length seven.

Limited sorters

We can impose bounds on the sizes of either the push or pop operations (or both). An interesting special case is when the push operation is limited to a single element, and the pop operation may or may not be so limited.

We would like to find generating functions for the classes in this case, i.e. expressions of the form:

$$\sum_{n=0}^{\infty} \left(\begin{array}{c} \text{number of sortable} \\ \text{sequences of length } n \end{array} \right) x^n.$$

Generating functions

Let f_k be the generating function for sortable sequences when the push operation is limited to one element, and the pop is limited to k.

In a sortable sequence, π, let t be the maximum for which the elements 1 through t occur in π in decreasing order. Then, the intervening blocks must be sortable, and the final pop of t elements must be allowed (modulo a minor quibble when 1 is the last element of π).

GF for $(1, k)$ sorting

So:

$$f_k = 1 + x f_k + (f_k - 1) \sum_{t=1}^{k} x^t f_k^t.$$

The different values of k provide a link from the Catalan numbers ($k = 1$) which satisfy a quadratic equation, through to a solution of a different quadratic ($k = \infty$).

Growth rates

The term controlling the exponential growth of the coefficients is the reciprocal of the radius of convergence. This increases from 4 at $k = 1$, to 5 at $k = \infty$. In fact, if we write:

$$c_k = 5 - e_k$$

then

$$e_k \approx \frac{C}{3^k}.$$

The moral is, *don't pay much for extra power once you can lift six plates.*

Other results

- If both the push and the pop operations are allowed to handle two or more objects then there are many sequences which can be sorted in more than one way.
- This complicates the enumeration problem enormously.
- For fixed bounds on the push and pop sizes we can prove: *There is a deterministic push down automaton whose accepted language contains precisely one sequence of operations to sort any sortable input sequence.*

Late breaking news

- More sequences can be sorted using pushes and pops limited to two elements at a time, than with pushes limited to one, and pops unlimited.
- The exponential growth rate for the $(2, 2)$ case is $5.412\ldots$, while that for the $(1, \infty)$ case is 5.
- The generating function $f_{2,2}$ for the $(2, 2)$ case satisfies an algebraic equation of degree 4, whose coefficients are polynomials in x of degree up to 4.

Thank you!