THE CYCLIC TOWERS OF HANOI

M.D. ATKINSON

Department of Mathematics and Statistics, Carleton University, Ottawa, Canada K1S 5B6

Received May 1981; revised version received August 1981

Towers of Hanoi, recursion, iteration

The famous Towers of Hanoi puzzle consists of 3 pegs (A, B, C) on one of which (A) are stacked n rings of different sizes, each ring resting on a larger ring. The objective is to move the n rings one by one until they are all stacked on another peg (B) in such a way that no ring is ever placed on a smaller ring; the other peg (C) can be used as workspace. The problem has long been a favourite in programming courses as one which admits a concise recursive solution. This solution hinges on the observation that, when the largest ring is moved from A to B, the n - 1 remaining rings must all be on peg C. This immediately leads to the recursive procedure

```
{move n rings from peg x to peg y using peg z as workspace}
procedure hanoi(n, x, y, z: integer);
begin
    if n < > 0 then
    begin hanoi(n - 1, x, z, y),
        writeln('move a ring from peg, x, 't) peg', y);
        hanoi(n - 1, z, y, x)
    end
end
```

which (apart from: the extra begin-end imposed by **PASCAL** syntax) is about as brief as one could wish for.

Unfortunately for the teachers of recursion, several authors, for example [1,2], have noticed that this problem can be 'cooked' by an iterative solution which is hardly more complicated. On odd moves the smallest ring is always moved in some fixed direction (clockwise $A \rightarrow B \rightarrow C \rightarrow A$, or anticlockwise) while on even moves the only possible move with a larger ring is made. The lessons that one can draw from these two solutions are discussed in [2].

However it would be a pity to banish this attractive toy from the realm of recursive programming and I propose a small modification to the ground rules to make a problem whose most natural solution provides a pleasing example of two mutually recursive procedures; and which does not seem to be so easy to solve by iteration. The modification is simply to regard the 3 pegs as being arranged cyclically and to allow moves in the clockwise direction only. Then moving the n rings one position clockwise is not equivalent to moving them one position anticlockwise. These two distinct problems can be solved by interdependent recursive procedures. Again the key is to consider interim positions when the largest ring is moved and to realise that these positions have to be preceded and succeeded by solutions of corresponding problems with n-1 rings. Disregarding forward reference considerations the resulting procedures are

```
{move n rings one position clockwise from peg x to peg y
using peg z as workspace}
procedure clock (n, x, y, z: integer);
begin
    if n > 0 then
    begin anti(n - 1, x, z, y);
    writeln('move a ring from peg', x);
    anti(n - 1, z, y, x)
    end
end {clock};
{move n rings one position anticlockwise from peg X to peg
```

Y using peg Z as workspace} procedure anti(n, X, Y, Z: integer); begin if n > 0 then • •

If we let c_n and a_n denote the number of moves required to transfer the n rings one position clockwise and one position anticlockwise then it is evident that

$$c_{n} = \begin{cases} 0 & \text{if } n = 0, \\ 2a_{n-1} + 1 & \text{if } n > 0, \end{cases}$$
$$a_{n} = \begin{cases} 0 & \text{if } n = 0, \\ 2a_{n-1} + c_{n-1} + 2 & \text{if } n > 0. \end{cases}$$

These equations are easily seen to have the solution

$$c_{n} = \frac{1}{2\sqrt{3}} \left\{ (1 + \sqrt{3})^{n+1} - (1 - \sqrt{3})^{n+1} \right\} - 1 ,$$
$$a_{n} = \frac{1}{4\sqrt{3}} \left\{ (1 + \sqrt{3})^{n+2} - (1 - \sqrt{3})^{n+2} \right\} - 1 ,$$

both of which are $O(2.733^n)$ and can be compared with $2^n - 1$, the number of moves required to solve the ordinary Tower of Hanoi problem.

The procedures above print out the sequence of peg names that are the source of moves. However each move can be characterized also by the name of the ring being moved. It was pointed out to me by D.T. Barnard that one can modify the procedures so that they print out the sequence of names of rings being moved - just suppress the references to peg names and change the writeln statements to writeln('move ring', n).

References

- [1] P. Buneman and L. Levy, The towers of Hanoi problem, Information Processing Lett. 10 (4,5) (1980) 243-244.
- [2] P.J. Hayes, A note on the towers of Hanoi problem, Comput. J. (1977) 282-285.