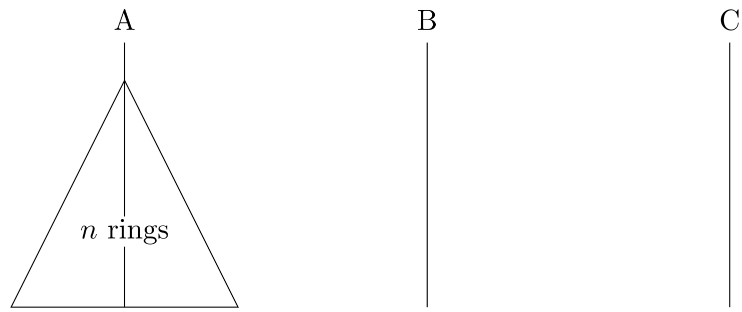
Lecture Notes, Fundamental Algorithms: Thinking Recursively, Part 1

In this note, we show some recursive solutions to Tower of Hanoi problems.

Tower of Hanoi

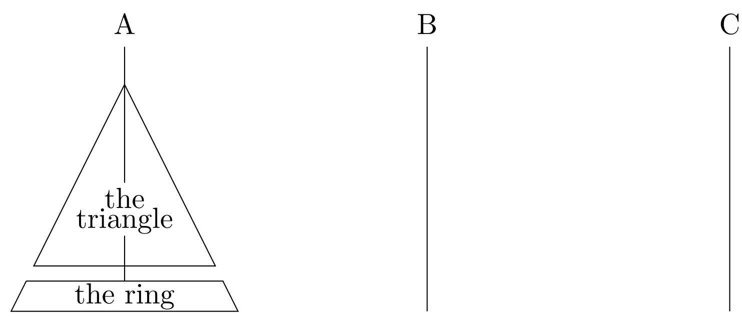
In the basic Tower of Hanoi problem there are three poles and n rings, with the i-th ring having a radius of i units (e.g. centimeters). The rule is that the rings on each pole, going from the bottom to the top have decreasing radii. We names the poles A, B, and C. In the standard version, we start with all n rings on A and the task is to move all the rings to pole B, always moving one ring at a time, and maintaining the rule about the rings from bottom to top on each pole having decreasing radii.

Below, we illustrate the initial configuration.



The base case has 1 ring, $n = 1$. In this case the solution is to move the single ring from its start location on A to its final location on B.

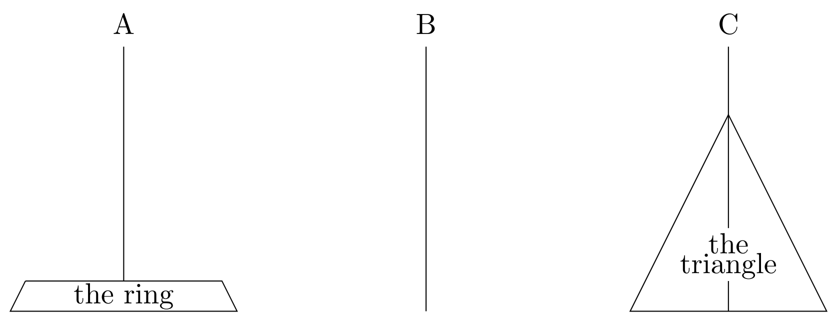
Before solving the general problem, we consider the following related ring and triangle problem. The input consists of two items, a ring and a triangle on pole A, with the triangle on top. They both need to be moved to pole B, and the ring may never be placed on top of the triangle.



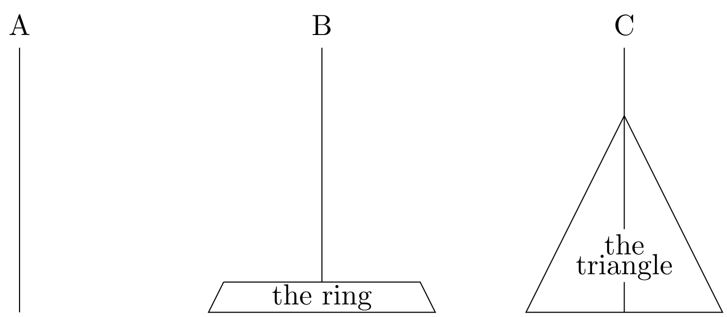
Here is the solution.

We are going to need to move the ring from A to B. So the first step is to get the triangle out of the way, i.e. move it to pole C.

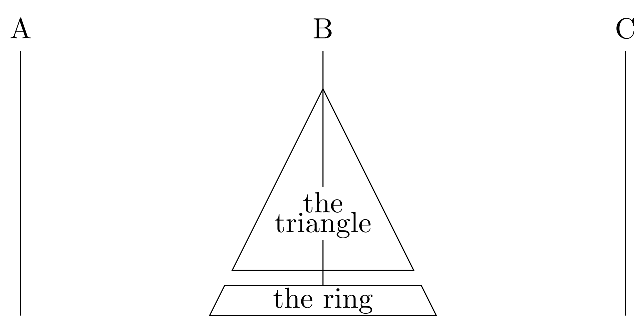
1. Recursively more the top $n-1$ rings out of the way (i.e. to pole C).



2. Now move the ring from A to B.



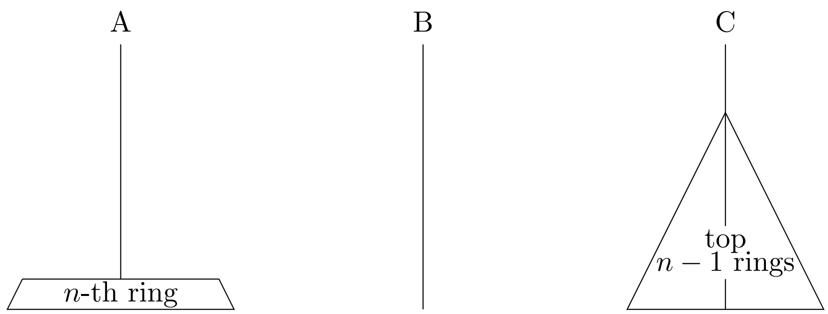
3. Finally, move the triangle from C to B.



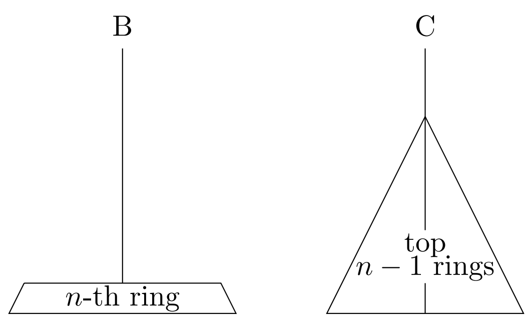
Now we return to the problem with n rings. All that changes is that we treat the top $n-1$ rings recursively, namely as if they formed the triangle. Here is the recursive solution.

We are going to need to move the n-th ring from A to B. So the first step is to get the other rings out of the way.

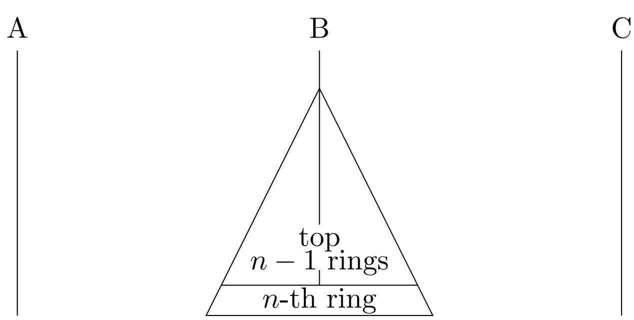
1. Recursively more the top $n-1$ rings out of the way (i.e. to pole C).



2. Now move the n-th ring from A to B.



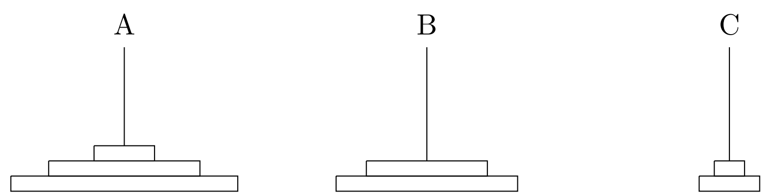
3. Finally, recursively move the top $n-1$ rings from C to B.



The base case, when $n=1$, consists of Step 2 alone.

We also remark that the recursive invocations of our algorithm are acting on the topmost k rings, for various $k<n$. In effect, the moves are occurring on top of the largest $n-k$ rings.

Tower of Hanoi with Disordered Start Here, we suppose that initially the rings are scattered across the poles (specified in an array Pos$[1:n]$, where Pos$[i]$ specifies the pole on which ring i is located). For example:



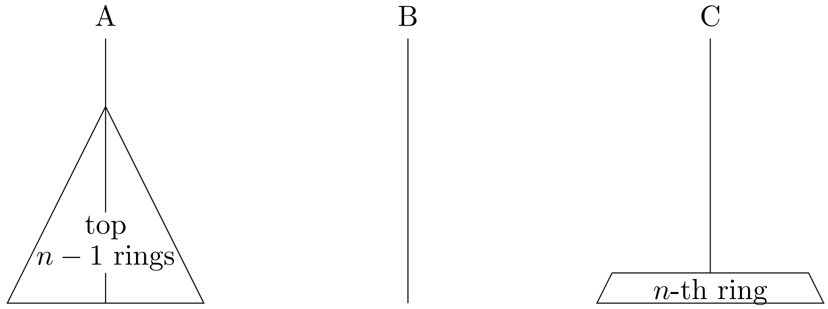
Let's suppose the goal is to get all the rings on Pole B. How to approach this? Again, we want to think recursively.

Let's think about the move ring n needs to make eventually. Suppose it starts on pole X. If $X = B$ then it is already at its destination. Otherwise, what should we do? Please try to come up with a solution before going onto the next page.

As before, we need to move the other rings out of the way so that ring n can be moved from pole X to pole B. We then complete the solution by moving the top $n-1$ rings to pole B.

By way of example, suppose that $X = C$. Then, we have the following 3 steps.

1. Recursively move the $n-1$ smallest rings to Pole A.



2. Move ring n to Pole B.
3. Recursively move the $n-1$ smallest rings to Pole B.

Let's also see this in program form. We will call the procedure DisToH. It will take three arguments, $k \le n$, the number of rings to move, Pos$[1:n]$, the array holding the current positions of the rings, and D, the pole to which the top k rings are to be moved.

It will be useful to have a function Thrd(X, Y) which returns the pole $Z \ne X, Y$.

DisToH(k, Pos, D)
    if $k = 1$ **and** Pos$[k] \ne D$ **then** move ring 1 to Pole D
    else if Pos$(k) = D$ **then** DisToH$(k-1, \text{Pos}, D)$
        else do
            $E \leftarrow$ Thrd(Pos$(k), D)$;
            DisToH$(k-1, \text{Pos}, E)$;
            move ring k to Pole D; Pos$[k] \leftarrow D$;
            DisToH$(k-1, \text{Pos}, D)$
        end (* else do *)
end (* DisToH *)

The inductive assertion is that DisToH(n, Pos, D) moves the top n rings from their initial positions to Pole D using legal moves. This is readily verified.