

Charlton Lin

Professor Bai

Great Ideas in Computer Science

13, March, 2013

Tower of Hanoi

The Tower of Hanoi is a 19th century European game of 3 pegs and a set number of disks (64 in the original game) stacked on one peg, each disk smaller than the one under it. The object of this game is to move the entire stack of disks from one peg to another. In some versions of the game, the stack is to move from the leftmost peg to the middle, while in other versions the stack is to move from the leftmost peg to the rightmost. Nevertheless, the rules are that the towers of disks can only be moved one disk at a time, and no disk may rest on another disk that is smaller than itself. In this essay, the disks must be moved from the leftmost peg to the rightmost.

The shortest way to solve the Tower of Hanoi with n disks is to first move the top $n-1$ disks to the middle peg (a temporary peg), then move the final (largest) disk to the rightmost peg (the destination), and move the remaining disks from the middle peg (the temporary peg) to the rightmost peg (the destination). Although this algorithm goes against the rules, we can use it to simplify each step of the problem. If we cut the problem up into smaller towers, we can find an easy recursive algorithm to solve the puzzle.

The recursive solution requires a base case. If there is only one disk to move, the disk can be moved from its origin to the destination. After this is done, the recursion can end. The next part of the recursive function is the steps. The solution proposed above can act as the steps: move the top $n-1$ disks to a temporary tower, then move the leftover disk to its destination, then move the disks in the temporary tower to the destination.

```
hanoi(n, fromTower, toTower, tempTower) //moves n disks from [fromTower] to [toTower]
if n == 1: //Base Case
    {move disk from tower [fromTower] to tower [toTower]}
else: //Recursive Steps
    hanoi(n-1, fromTower, tempTower, toTower)
    hanoi(1, fromTower, toTower, tempTower)
    hanoi(n-1, tempTower, toTower, fromTower)
```

The above recursive function hanoi, written in pseudo-code, represents the solution for any number of disks for the Tower of Hanoi. If the base case is reached, the tower is moved, otherwise, the recursive steps break the problem down into smaller problems that can be solved with the same solution. The function is recursive because it calls itself within the function. Because the Tower of Hanoi can be broken down into smaller problems, and each smaller problem is closer to the solution, the puzzle can be solved recursively.

An interesting twist to the Tower of Hanoi is the Reve's Puzzle. The game has the same setup and rules except there are 4 pegs instead of 3. While the 4 pegs makes the game easier to solve, and the function above can still be used to solve the puzzle, it is no longer the most efficient solution. In fact, to this day, algorithms used to solve Reve's Puzzle have not been proven to be the most efficient. For large numbers of disks, the shortest number of moves required to solve Reve's Puzzle is unknown. This puzzle is an example of how one small change to an otherwise easily recursively solved problem can make the problem extremely difficult to solve.

Bibliography

Rhoads, S. E. (2012, September 14). *The Reve's Puzzle*. Retrieved March 12, 2013, from

Wolfram Demonstrations Project: <http://demonstrations.wolfram.com/TheRevesPuzzle/>

Roberts, E. S. (1986). *Thinking Recursively*. John Wiley & Sons, Inc.

Tower of Hanoi. (n.d.). Retrieved March 12, 2013, from Maze Works:

<http://www.mazeworks.com/hanoi/index.htm>

Weisstein, E. W. (2013). *Tower of Hanoi*. Retrieved March 11, 2013, from Mathworld--A

Wolfram Web Resource: <http://mathworld.wolfram.com/TowerofHanoi.html>