Case Western Reserve University
Est. 1826

Recursion

Prof. Dianne Foreback

Information on slides originating from *Algorithm Design & Applications* by Michael T. Goodrich and Roberto Tamassia, © 2015 John Wiley & Sons, Inc.Goodrich and Tamassia, ISBN: 978-1118335918.

Sites

- Animation of Towers of Hanoi
  https://www.mathsisfun.com/games/towerofhanoi.html

Recursive Algorithm

- A recursive algorithm is an algorithm that invokes itself

Rules of Recursion

- Requires a **base case(s)** - value for which function is directly known without resorting to recursion
- Make **progress** – recursive call must always make progress to a base case
- **Compound interest rule** – never duplicate work by solving the same instance of a problem in separate recursive calls

Iterative Factorial Algorithm

Recall: $n! = n(n-1)(n-2)...2*1$

$0! = 1$

Algorithm: FactorialIterative (n)

Input: $n \geq 0$ and n is an integer

Output: $n!$

```
fact ← 1
for j ← 2 to n do
        fact ← fact * j
return j
```

Recursive Factorial Algorithm

Recall: $n! = n(n-1)(n-2)...2*1$

$0! = 1$

Algorithm: FactorialRecursive (n)

Input: $n \geq 0$ and n is an integer

Output: $n!$

```
if n ≤ 1
        return 1
else
        return n * FactorialRecursive (n-1)
```

Divide & Conquer Approach

- Algorithm divides a problem into subproblems
- solves each subproblem recursively
- puts together the partial solution eventually solving the total solution

Towers of Hanoi

- **Goal**: Move all disks to destination peg
- **Rules**:
  - Only 1 disk can be moved at a time
  - A disk can be moved to any peg providing rule 3 holds
  - Larger disks cannot be placed on top of smaller disks

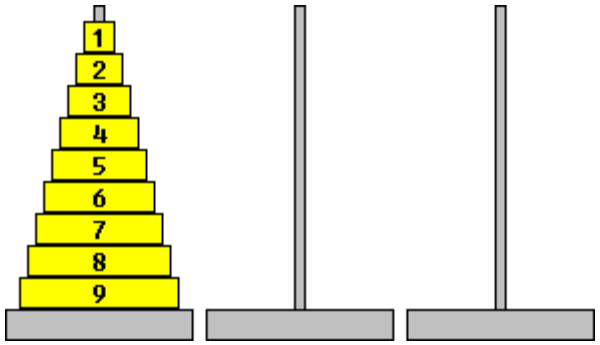

Image from https://www.jaapsch.net/puzzles/hanoi.htm

View Animation: Towers of Hanoi

- Animation of Towers of Hanoi
  https://www.mathsisfun.com/games/towerofhanoi.html

Algorithm Towers of Hanoi

Algorithm: Hanoi (n, start, dest, temp)
Input: number of disks. n > 0, start, dest, temp as integers each with a unique value of 1, 2, or 3
Output: Sequence of moves to solve the Towers of Hanoi problem

```
if n = 1
  print "move disk from" start "to" dest
else
  Hanoi (n-1, start, temp, dest)
  print "move disk from" start "to" dest
  Hanoi (n-1, temp, dest, start)
```

Thank You !

Questions ?