

Recursion

Recursive Algorithms

C1





Australian
National
University

A vintage-style advertisement for Droste Cocoa. The background is a solid, vibrant red. At the top center is a royal coat of arms. Below it, the word "Droste" is written in a large, elegant, gold-colored script font with a registered trademark symbol. Underneath the brand name, the word "HOLLAND" is printed in a smaller, gold, sans-serif font. The central figure is a woman dressed as a nun in a white habit and a dark brown robe with white cuffs. She is holding a silver tray with both hands. On the tray are a box of Droste Cocoa and a white mug. The box is red and white, featuring the brand name and the nun's image. The mug is white with a small illustration of the nun. At the bottom of the advertisement, the word "cocoa" is written in a large, bold, gold, sans-serif font. Below that, the text "NET WT 250g e 8.8 oz." is printed in a smaller, gold, sans-serif font. At the very bottom left, the word "Kosher" is written in a small, gold, sans-serif font, followed by a U-shaped symbol.

cocoa

NET WT 250g e 8.8 oz.

Kosher U

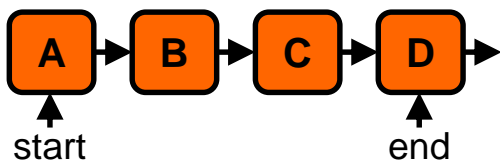


Austr
Natio
Unive

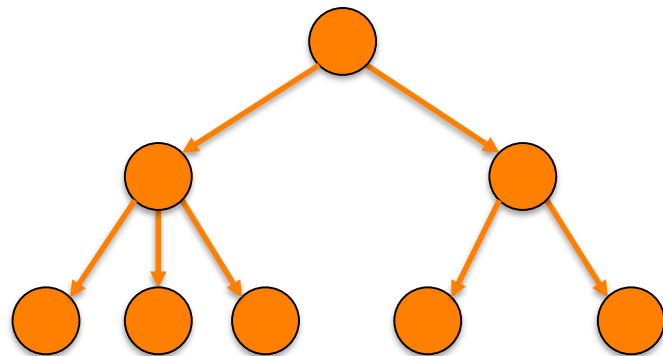


Recursive Data Structure

A recursive data structure is comprised of components that reference other components of the same type.



linked list



tree

Recursive Algorithms

A recursive algorithm references itself.

A recursive algorithm is comprised of:

- one or more base cases
- a remainder that reduces to the base case/s

Example: Fibonacci sequence

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377...

$\text{fib}(0) = 1$ (*base case*)

$\text{fib}(1) = 1$ (*base case*)

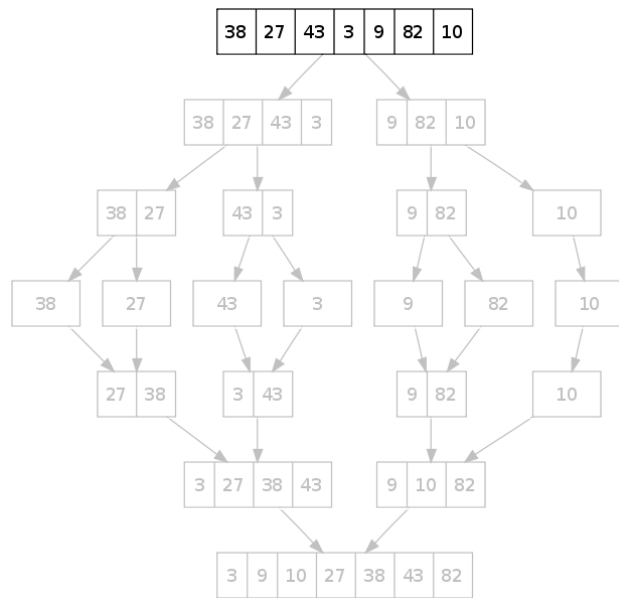
$\text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2)$ (*for* $n \geq 2$)



Example: Mergesort (von Neumann, 1945)

Sort a list

- List of size 1 (*base case*)
 - Already sorted
- List of size > 1
 - Split into two sub lists
 - Sort each sub list (*recursion*)
 - Merge the two sorted sub lists into one sorted list (by iteratively picking the lower of the two least elements)



Animation: Visualizing Algorithms, Mike Bostock, bost.ocks.org/mike/algorithms