Recursion

Recursive Algorithms
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…

\[
\begin{align*}
\text{fib}(0) &= 1 \quad (\text{base case}) \\
\text{fib}(1) &= 1 \quad (\text{base case}) \\
\text{fib}(n) &= \text{fib}(n-1) + \text{fib}(n-2) \quad (\text{for } n \geq 2)
\end{align*}
\]
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
Example: Mergesort (von Neumann, 1945)