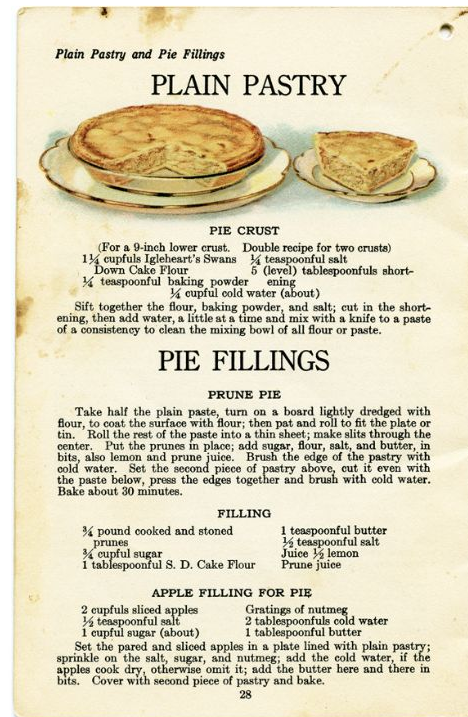


Introduction to Algorithms

Algorithms

Basic definition: A sequence of steps that solves a problem or performs a task



Algorithms

This week you'll learn:

Several programming algorithms

Big O Notation

Algorithm techniques: Brute force

Simplest technique, trial and error

Evaluate all options to find the potential solution

Example: Try all 10,000 possibilities for 4-digit PIN

Inefficient, relies heavily on computing power

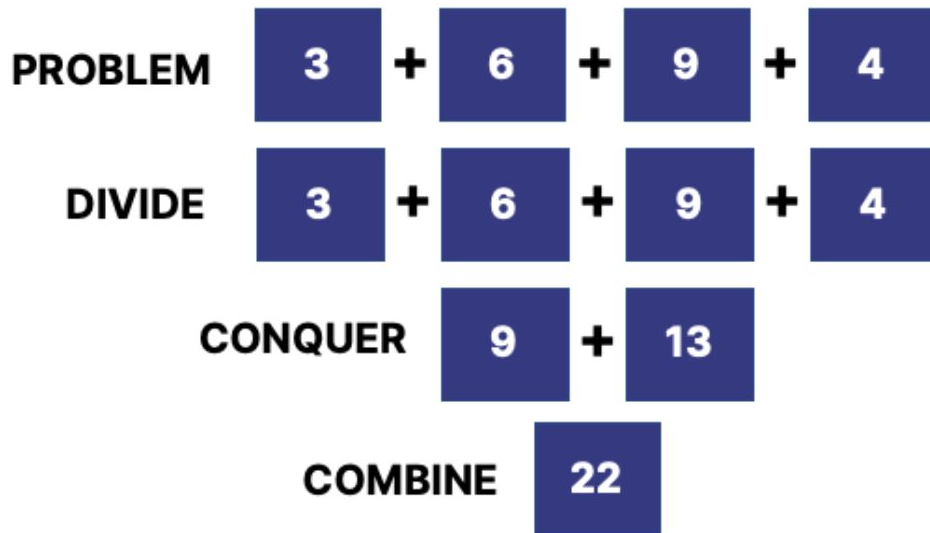
Algorithm techniques: Divide & Conquer

Divide: Divide problem into two sub-problems

Conquer: Solve sub-problems recursively

Combine: Combine sub-problem solutions together

Algorithm techniques: Divide & Conquer



More algorithm techniques

Recursion: Function calls itself until base case is met

Dynamic Programming: Similar to Divide & Conquer, uses memoization

Greedy: Builds solution step by step,
chooses step with most immediate benefit

Backtracking; branch and bound; and others

Searching & sorting algorithms

Widely used – we must have ways to search and sort computerized data

For every search or sort task, we must understand how to do it efficiently

The wrong algorithm may perform the task, but waste time and resources

Many algorithms – this week gives you a starting point

Get excited! Algorithms are powerful and interesting

Searching algorithms

Linear search:

Easy to understand

Will always work, regardless of how items are sorted... eventually

Binary search:

Items must be sorted

More efficient than linear search

Sorting algorithms

Being able to sort data sets is very important

Can use other, more efficient algorithms on sorted set, e.g. binary search

Bubble sort

Uses brute force, less efficient

Quicksort

Uses divide & conquer, more efficient