Introduction: Course Overview

Daniel Kane

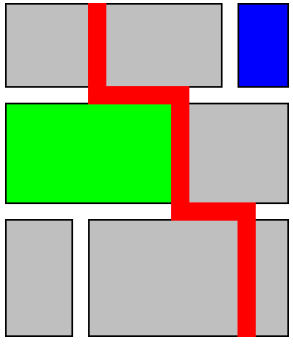
Department of Computer Science and Engineering
University of California, San Diego

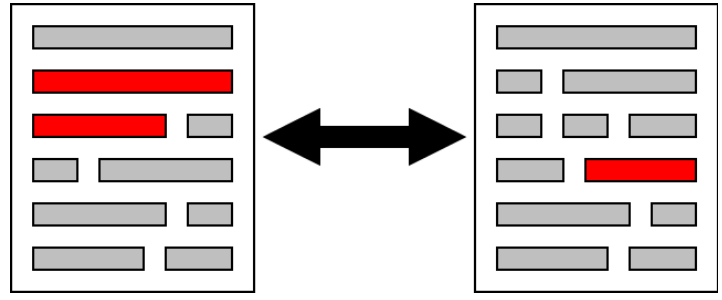
Algorithmic Design and Techniques
Algorithms and Data Structures

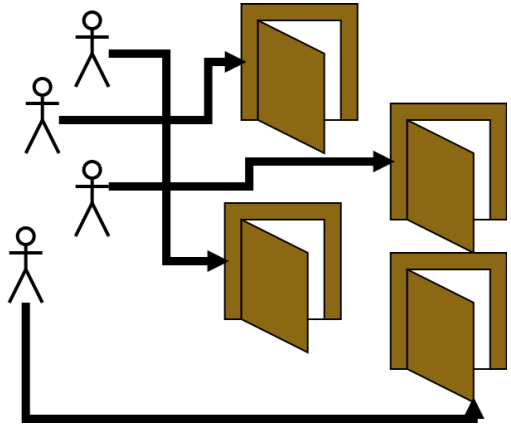
Algorithm Design is Hard

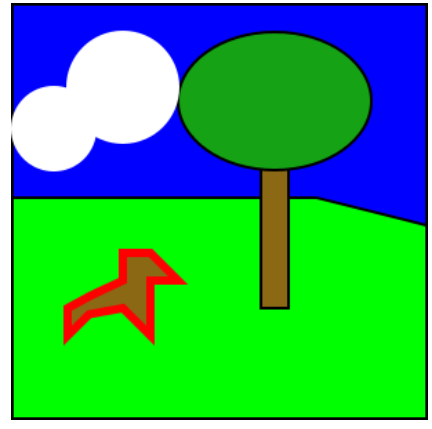
- Algorithms very general.
- No generic procedure for designing good algorithms.
- Finding good algorithms often requires coming up with unique insights.

Algorithms Solve Many Different Problems

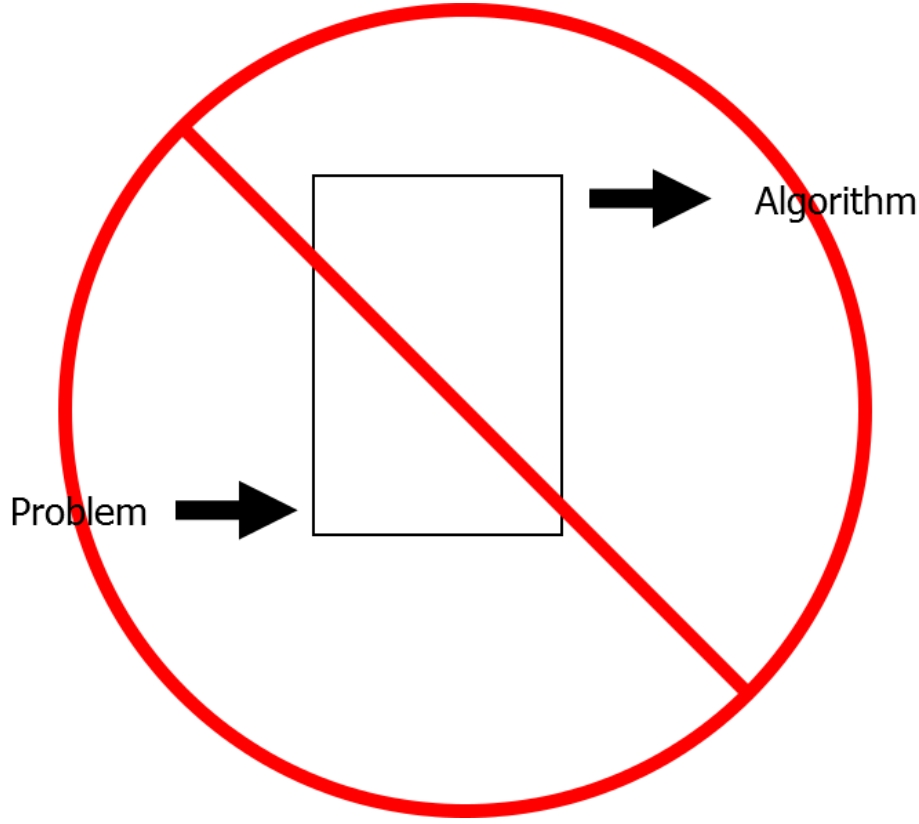




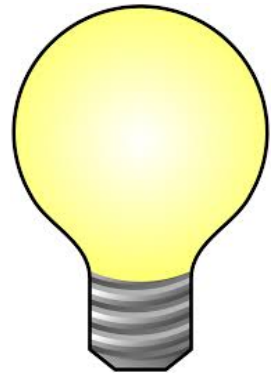




No Generic Procedure to Create Algorithms



Finding Algorithm Often Requires Unique Insights



Toolbox

What can we teach you?

- Practice designing algorithms.
- Common tools used in algorithm design.

Toolbox

What can we teach you?

- Practice designing algorithms.
- Common tools used in algorithm design.
- We will discuss three of the most common algorithmic design techniques:
  - Greedy Algorithms
  - Divide and Conquer
  - Dynamic Programming

Levels of Design

Naive Algorithm: Definition to algorithm.
Slow.

Levels of Design

Naive Algorithm: Definition to algorithm.
Slow.

Algorithm by way of standard Tools:
Standard techniques.

Levels of Design

Naive Algorithm: Definition to algorithm.
Slow.

Algorithm by way of standard Tools:
Standard techniques.

Optimized Algorithm: Improve existing algorithm.

Levels of Design

Naive Algorithm: Definition to algorithm. Slow.

Algorithm by way of standard Tools: Standard techniques.

Optimized Algorithm: Improve existing algorithm.

Magic Algorithm: Unique insight.

The Rest of the Course

- Each unit covers a technique.
- Exercises help build intuition.