CS F364
Design & Analysis of Algorithms

Algorithm Design Techniques

Comparison:

Top-Down Design vs. Bottom-up Design

Divide-and-Conquer vs. Dynamic Programming

Top Down Design vs. Bottom Up Design

Top Down Design

- Strategy: Divide problem
  - Start: Given problem
  - End: Atomic problems
- Strength:
  - Easy to frame solutions for new problems
- Weakness:
  - Overlapping sub-problems may be solved more than once

Bottom Up Design

- Strategy: Combine solutions
  - Start: Atomic solutions
  - End: Solution to given problem
- Weakness:
  - Problem structure must be analyzed apriori
    - If not, what happens?
- Strength:
  - Every sub-problem is solved at most once.

Divide and Conquer vs. Dynamic Programming

Top Down Design

- Divide and Conquer is a special case of Top Down Design:
  - sub problems have the same structure as the given problem

Bottom Up Design

- Dynamic Programming is a special case of Bottom Up Design:
  - Solution to the given problem has the same structure as those of the sub problems