Dynamic Programming
Chapter 15
Rod Cutting Problem
Rod Cutting Problem

- Given a rod of length $n$ inches and a table of prices $p_i$ for $i = 1, 2, \ldots, n$, determine the maximum revenue $r_n$ obtainable by cutting up the rod and selling the pieces.

- Naïve solution?
  - Try all possibilities. Exponential!
Recursive Solution

Rod-Cutting(n, p)

if n = 0 then return 0
best-cut = p[n]
for cut = 0 to n
  value = Rod-Cutting(cut) + Rod-Cutting(n – cut)
  best-cut = max(best-cut, value)
return best-cut
Memoized Recursive Solution

- Rod-Cutting(n, p)
  - if n = 0 then return 0
  - if table.contains(n) then return table[n]
  - best-cut = p[n]
  - for cut = 0 to n
    - value = Rod-Cutting(cut) + Rod-Cutting(n – cut)
    - best-cut = max(best-cut, value)
  - return best-cut
Bottom-up Iterative Solution

Rad-Cutting(n, p)

- table = {}
- if n = 0 then return 0
- for i = 1 to n
  - best-cut = p[i]
  - for j = 1 to i-1
    - value = max(value, table[j], table[i-j])
  - table[i] = best-cut
- return table[n]
Dynamic Programming

- You have a big problem
- You can break it down into smaller subproblems
- You don’t know the best way to split it
  - So, you try all possibilities
- Identify similar subproblems that are solved many times
- Devise a better (polynomial) algorithm