CSCE 750 | Dynamic Programming (cont.)

10/12/2023

Idea: Given a problem:

1) Identify a family of subproblems that are related. (The original problem may or may not be one of these.)
2) Show how the original problem can be solved easily given solutions to one or more subproblems.
3) Identify base case
4) Solve subproblems from the bottom-up, starting with base cases, then solving new subproblems in terms of previously found solutions.

~~Ex: Connecting wires. Given: a collection of wires with various integer lengths. Each wire has a female (F) plug on one end and a male (M) plug on the other. Can plug an F & M together to make a longer wire, but only if they are the same type [Three types: A, B, C]~~

Rod-Cutting

Given: Rod of length ℓ [all lengths are integers]

~~Given~~ Price array $P[1..\ell]$ where

arbitrary nonnegative [$P[j]$ is the price you can get for selling a rod of length j $(1 \le j \le n)$

Find ~~an opt~~ a way to cut the rods to maximize the total revenue (sum of the prices).

Let $R(k) :=$ max revenue starting with a rod of length k. $(0 \le k \le \ell)$

[Want is $R(\ell)$.]

$\rightarrow$ $R(0) = 0$

$k > 0$

can get $R(k-i) + P[i]$

= best revenue <u>assuming</u> the rightmost rod after cutting has length i.

$$\rightarrow R(k) := \max_{i:=1}^{k} \{ R(k-i) + P[i] \}$$

Table $R[0..\ell]$

$R[0] := 0$

for $k := 1$ to ℓ do

$$\begin{array}{l} R[k] := \max\{R[k-i] + P[i] : 1 \le i \le k\} \\ S[k] := \operatorname{argmax}\{\ \text{- - - - -}\ \} \end{array}$$

return $R[\ell]$

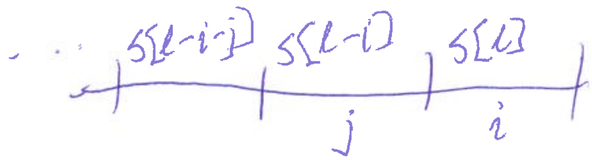
Want to find an optimal cutting, need to augment with more info:

To get the cuts:

while $\ell > 0$:

  output $S[\ell]$

  $\ell := \ell - S[\ell]$



Matrix Chain Order Given a list $A_1, A_2, \dots, A_n$ of matrices such that ~~there~~ their product $A_1 A_2 \cdots A_n$ is well-defined

let $p_0, \dots, p_n$ be positive integers such that

A_k is a $p_{k-1} \times p_k$ matrix

$$\begin{array}{ccccc} A_1 & A_2 & A_3 & \cdots & A_n \\ p_0 \times p_1 & p_1 \times p_2 & p_2 \times p_3 & & p_{n-1} \times p_n \end{array}$$

Matrix mult is associative, but some groupings are more efficient than others.

Ex: $A = 10 \times 10$

$B = 10 \times 15$

$C = 15 \times 2$

$10 \times 10 \times 15 + 10 \cdot 15 \cdot 2$

$(\underbrace{AB}_{10 \times 15})C$ takes 1800 multiplications

$\underbrace{A(BC)}_{200 + 300}$ takes 500 multiplications

$300 = 10 \cdot 15 \cdot 2$

Multiplying an $m \times r$ matrix with an $r \times n$ matrix takes mrn many multiplications and results in an $m \times n$ matrix

Problem: Given matrices $A_1, \ldots, A_n$ and sizes $p_0, \ldots, p_n$ what is the least # of scalar multiplications needed to compute $A_1 \cdots A_n$?

[For simplicity, only count multiplications, as they dominate all other steps of the matrix mult algorithm.]

Notation: for $1 \leq i \leq j \leq n$, let $A_{i..j}$ be the product of $A_i \cdots A_j$ (a $p_{i-1} \times p_j$ matrix).

Subproblem: least # of multiplications to ~~find~~ compute $A_{i..j}$

Let $M[1..n, 1..n]$ be such that for $1 \le i \le j \le n$ (G

$M[i,j]$ = least # of mults to compute $A_{i..j}$

$$(A_1 \quad A_{k-1})(A_k \quad A_n)$$

last mult in an optimal grouping,
for some k, where $A_{1..(k-1)}$ and
$A_{k..n}$ are computed optimally.

(To be continued)