Chained Matrix Multiplication and Recursion

Lecture 16
CS 312

Project 2

• Times are strange
  – <30sec , <40sec, <50sec, <60sec, >60sec
• Sample data sets and solutions available

Resources

• “Writing For Computer Science” by Justin Zobel
• “The Visual Display of Quantitative Information” by Edward R. Tufte
• Writing center 1010 JKHB, 8-4306

Project 1

• Experiments were, in general, very interesting
• Do not narrate: “First, I did ... Then I did .... Then I did ... ”
• I like graphs more than tables for quantitative information
• Follow the guidelines on the webpage:
  – dbl space, cover sheet
• Section headings don’t hurt
• Lengths were good

Project 3

• Don’t start project 3 yet. It might change

Objective

• Optimize chained matrix mult. using DP (bottom-up first, then top-down)
• Rewrite bottom-up DP algorithms as top-down algorithms using a table
Bottom-up matrix mult.

for \( s = 0 \) to \( n - 1 \)
if \( s = 0 \) then for \( i = 1 \) to \( n \) \( m[i,i] := 0 \)
else
  if \( s = 1 \) then for \( i = 1 \) to \( n-1 \) \( m[i,i+1] := d[i-1] \cdot d[i] \cdot d[i+1] \)
  else
    for \( i = 1 \) to \( n - s \)
      \( m[i,i+s] := \text{infinity} \)
    for \( k = i \) to \( i + s \)
      \( m[i,i+s] := \min(m[i,i+s], m[i,k] + m[k+1,i+s] + d[i-1] \cdot d[k] \cdot d[i+s]) \)
    \( m[i,i+s] := \text{so far} \)

Bottom-up vs. Top-down

- Might compute irrelevant subsolutions
- Manage recursion

Top-down Recursive Approach

function \( fm(i,j) \)
if \( i = j \) then return 0
\( m := \text{infinity} \)
for \( k = 1 \) to \( j - 1 \) do
  \( m := \min(m, fm(i,k) + fm(k+1,j) + d[i-1] \cdot d[k] \cdot d[j]) \)
return \( m \)

What's the complexity of this algorithm?

Call Tree

function \( fm-mem(i,j) \)
if \( i = j \) then return 0
if \( \text{mtab}[i,j] > -1 \) then return \( \text{mtab}[i,j] \)
\( m := \text{infinity} \)
for \( k = 1 \) to \( j - 1 \) do
  \( m := \min(m, fm-mem(i,k) + fm-mem(k+1,j) + d[i-1] \cdot d[k] \cdot d[j]) \)
\( \text{mtab}[i,j] := m \)
return \( m \)

How do you modify \( fm \) to avoid recomputing results?

Memory Function

Call Tree
Call Tree

Homework

• Problem 8.27. Replace “and” with “or”
  – knapsack is the project though.
• Fill in the table using a memory function