CSE 102 11-9-21

- mid1 graded
- mid2 Thur. NOV. 18
- holiday Thur NOV. 11 (SUPP. lecture)
- hw5 : extended 1 day
- hw6 : soon

note:

Longest Path Problem in a Graph does not exhibit optimal substructure.

How to find a dynamic programming soln. to an opt. problem.

1.) charactize an opt. solution as a comb. of opt. subinstance solns. show that solving an instance involves one or more choices that leave subinstances of same problem.

2.) recursively define an opt. soln in terms of opt. subinstance solns. i.e. write down a recurrence relation.

3.) compute value of an opt. soln. in a bottom-up fashion, i.e. fill in the table.

4.) Construct an opt. soln. by backtracking through table in (3).

Problem: Matrix chain multiplication

Consider multiplying 2 matrices

A of size $p \times q$

B " " $q \times r$

Product $A \cdot B$ has size $p \times r$. Its ij^{th} is

$$\sum_{k=1}^{q} a_{ik} b_{kj},$$

which involves q multiplications.

Total # of multiplications $= p \cdot q \cdot r$

Now consider 3 matrices A, B, C of sizes

$A : p \times q$

$B : q \times r$

$C : r \times s$

we have 2 Parenthesizations of Product ABC.

(1) $\underbrace{A}_{p\times q} \cdot (\underbrace{B \cdot C}_{q\times s})$ cost $= pqs + qrs$

(2) $(\underbrace{A \cdot B}_{p\times r}) \cdot \underbrace{C}_{r\times s}$ cost $= pqr + prs$

Ex. $p = 10, q = 100, r = 10, s = 100$. Check

(1) cost $= 200{,}000$

(2) cost $= 20{,}000$

Exercise

- write the 5 parenthesizations of prod. of 4 matrices.
- determine # of paren. of 5 matrices and write them down.

Exercise

let $P(n)$ be the # of parens. of n an n-matrix product: $A_1 A_2 \cdots A_n$.

Show that

$$P(n) = \begin{cases} 1 & n = 1 \\ \sum_{k=1}^{n-1} P(k) \cdot P(n-k) & n \geq 2 \end{cases}$$

Difficult exercise:

Show $P(n) = \frac{1}{n} \binom{2n-2}{n-1}$

Exercise

show $P(n) = \Theta\left(\frac{4^n}{n^{3/2}}\right)$

note: $C_n = P(n+1)$ is called the Catalan numbers.

Problem MCM1

Given $n \geq 1$ and a matrix chain Prod. $A_1 A_2 \cdots A_n$ where A_i has size $p_{i-1} \times p_i$ $(1 \leq i \leq n)$, determine an optimal Parenthesization of the Prod. that minimizes ~~#~~ of multiplications.

note: Input to Problem is a vector

$(p_0, p_1, p_2, \ldots, p_n)$

Rmk.

as usual, 2 Problems

- find value of opt. soln, i.e. the min. # of mult. necessary
- find opt. soln. itself i.e. where to put Pa-entheses?

Look at general sub-instance of our Problem: multiply $(A_i A_{i+1} \cdots A_j)$ for some $1 \le i \le j \le n$. note any Paren. has a 'last' Product, i.e. an 'outermost' or 'top-level' Product.

$$(A_i \cdots A_k) \cdot (A_{k+1} \cdots A_j)$$

for some k in $i \le k < j$

Call k the top level split point.

claim

If $A_i \cdots A_j$ is optimally parenthesized, then so are both factors

$$(A_i \cdots A_k) \text{ and } (A_{k+1} \cdots A_j)$$

Proof:

Assume, to get a $\not\in$ that the Parenthesization of $(A_i \cdots A_k)$ is not optimal. Then we can replace that Paren. by another better (lower cost) Paren. of $(A_i \cdots A_k)$

But this gives us a better Paren. of the Prod. $(A_i \cdots A_k)\cdot(A_{k+1} \cdots A_j)$, which is impossible. ∎

<u>note</u> this argument shows us that, if we know the split point k, then opt. cost for $A_i \cdots A_j$ is a combination

$$\text{cost}(A_i \cdots A_j) = \text{cost}(A_i \cdots A_k) + \text{cost}(A_{k+1} \cdots A_j) + p_{i-1} p_k p_j$$

since

$$\underbrace{(A_i \cdots A_k)}_{p_{i-1} \times p_k} \cdot \underbrace{(A_{k+1} \cdots A_j)}_{p_k \times p_j}$$

Define

$m[i,j]$ = min # of mult. necessary to compute $A_i \cdots A_j$

$(1 \le i \le j \le n)$

note: $m[i,i] = 0$ (initialization)

goal: find $m[1,n]$

we've found that if k is the top level split point in an opt. paren., then

$$m[i,j] = m[i,k] + m[k+1,j] + p_{i-1} p_k p_j$$

Range of k: $i \le k < j$

Thus

$$m[i,j] = \begin{cases} 0 & (i = j) \\ \min\limits_{i \le k < j} \left(m[i,k] + m[k+1,j] + p_{i-1} p_k p_j \right) & (i < j) \end{cases}$$

note: $m[i,j]$ is undefined when $i > j$

See Pages 10 - 11 of handout on Dynamic Programming.