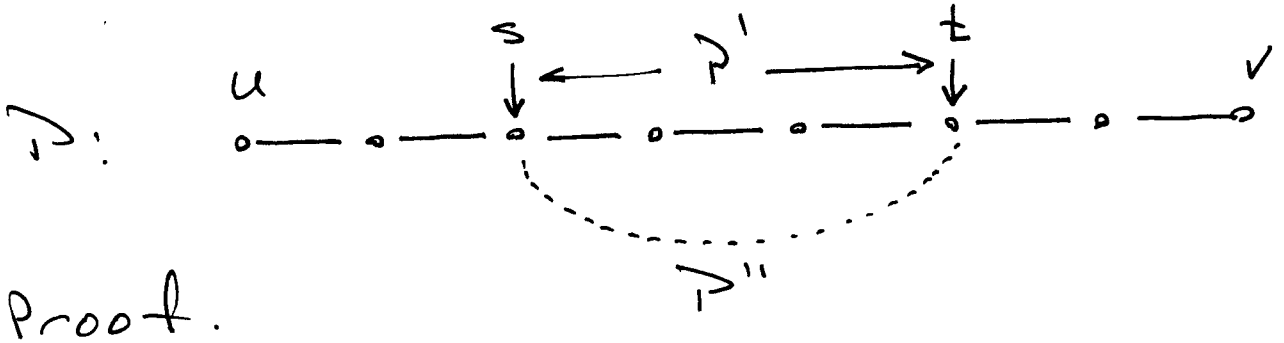
CSE 201 2-18-21

continue: shortest & longest Paths

Thm

Any subsequence of a shortest Path is also a shortest Path.



Proof.

Let P be a shortest u-v Path in G. Let s, t be any two intermediate vertices along P. Let P' be the sub Path of P joining s to t. must show P' is a shortest s-t Path.

Assume, to get a contradiction, that P' is not a shortest s-t Path. Then there must exist a shorter s-t Path in G; call it P''.

Let P''' be the u-v Path obtained by concatenating 3 Paths

- u to s along P : P_1
- s to t along P'' : P''
- t to v along P : P_2

Then

$$\begin{aligned} \text{length}(P''') &= \text{length}(P_1) + \text{length}(P'') + \text{length}(P_2) \\ &< \text{length}(P_1) + \text{length}(P') + \text{length}(P_2) \\ &= \text{length}(P), \end{aligned}$$

contradicting that P is a shortest u-v Path.

Therefore, no such shorter s-t Path P'' exists, and hence P' is a shortest s-t Path, as claimed.

■

what about longest Paths?

Ex. P.7 of Dyn. Prog. handout.

A sub Path of a longest Path is not necessarily a longest Path.

so

Problem 1: shortest Path satisfies Principle of optimality, and

Problem 2: longest Path does not.

Summary of Dyn. Prog. Procedure

1. characterize the structure of an opt. solution as consisting of optimal sub-instance solutions.
2. Recursively define an optimal soln. in terms of opt. subinstance solns, i.e. write a recurrence formula for value of opt. solution.
3. compute value of opt. solution in a bottom-up fashion, i.e. fill in a table giving values of sub-instance solns.
4. Construct an opt. soln. recursively by backtracking through table.

Problem: Matrix Chain multiplication

Defn a matrix chain is a product

$$A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdots A_{n-1} \cdot A_n$$

A_1: $p_0 \times p_1$, A_2: $p_1 \times p_2$, A_3: $p_2 \times p_3$, A_{n-1}: $p_{n-2} \times p_{n-1}$, A_n: $p_{n-1} \times p_n$

that is well defined, i.e. # col. in one matrix is eq. to # rows in next matrix.

Ex $n=3$: $ABC = A \cdot (B \cdot C) = (A \cdot B) \cdot C$

A: $p \times q$, B: $q \times r$, C: $r \times s$; $B \cdot C$: $q \times s$; $A \cdot B$: $p \times r$

$[\text{cost of } A \cdot B] = p \cdot q \cdot r$ real no. mults.

$[\text{cost of } B \cdot C] = q \cdot r \cdot s$ " " "

$[\text{cost of } A(BC)] = pqs + qrs$ (1)

$[\text{cost of } (A \cdot B) \cdot C] = pqr + prs$ (2)

Let $p = 10, q = 100, r = 10, s = 100$

(1) gives : 200,000

(2) gives : 20,000

Problem given a matrix chain of size n, find the optimal Parenthesization

Exercise

- write the 5 Paren.s. in case $n = 4$
- determine # of Paren.s. if $n = 5$, then write them.

Exercise.

Let $P(n)$ = # of Parens. of a matrix chain of len. n.

show $P(n)$ satisfies

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

check:

n	1	2	3	4	5	...
$P(n)$	1	1	2	5	?	

Defn $C_n = P(n+1)$ are called the Catalan numbers.

Fact: $P(n) = \frac{1}{n} \cdot \binom{2n-2}{n-1} = \frac{1}{n}\binom{2(n-1)}{(n-1)}$

Exercise (hard) Prove this

By Stirling's formula

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

↑ exponential.

so Brute force is intractable.

Dyn. Prog. approach

sub-problem: optimally Parenthesize

$$A_i \cdot \cdots \cdot A_j$$

for $1 \le i \le j \le n$.

Any Paren. of $A_i \cdots A_j$ splits into two chains

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

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

claim

If $A_i \cdots A_j$ is optimally Paren., then so are both factors $A_i \cdots A_k$ and $A_{k+1} \cdots A_j$

$$\underbrace{A_i \cdots A_j}_{\text{if opt.}} = \underbrace{(A_i \cdots A_k)}_{\text{opt.}} \cdot \underbrace{(A_{k+1} \cdots A_j)}_{\text{opt.}}$$

($p_{i-1} \times p_k$ and $p_k \times p_j$)

then →

Proof.

Assume, to get a $\lightning$, that $A_i \cdots A_k$ is not optimally Parenthesized. Then we can Replace the Paren. of $A_i \cdots A_k$ by a better one, yielding a Paren. of $A_i \cdots A_j$ better than optimal $\lightning$.

$\therefore$ $A_i \cdots A_k$ must be opt. Parenthesized.

Similarly for $A_{k+1} \cdots A_j$. ∎

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

$m[i,j]$ = min cost of a Parenthesization of $A_i \cdots A_j$

goal find $m[1,n]$

we see from Prev. Proof:

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

we dont know what k is, so

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