CSE 102 2-18-20

Defn

Let G be a graph, $u, v \in V(G)$.

a u-v Path is a sequence of alternating vertices & edges

P: $u = x_0, e_1, x_1, e_2, x_3, \ldots, e_k, x_k = v$

where the x's are no repeated edges, & no repeated vertices (unless $u = v$, called a closed path.)

$\text{length}(P) = k = \#\text{edges}$.

notation

$d(u,v)$ = length of a min-length u-v Path

$\ell(u,v)$ = length of a max-length (longest) u-v Path.

Problem 1

Given G, $u,v \in V(G)$: determine $d(u,v)$ and a Path of length $d(u,v)$.

Problem 2

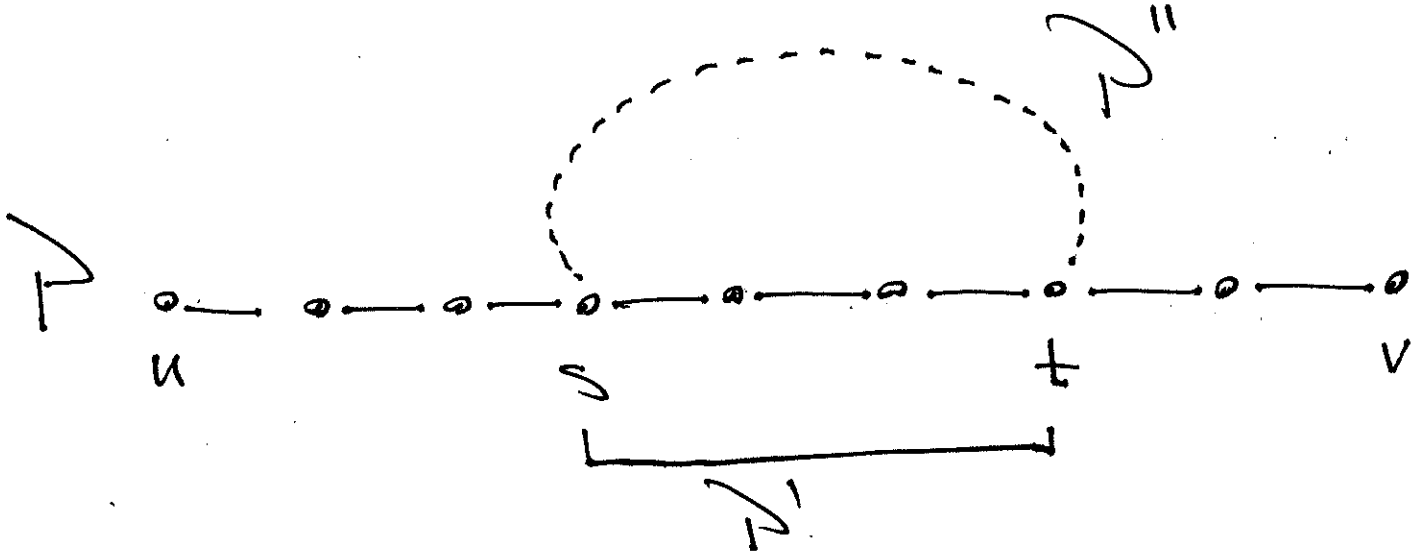
Given G, $u,v \in V(G)$: determine $\ell(u,v)$ and a Path of length $\ell(u,v)$.

Claim

Any subsequence of a shortest path is also a shortest path.
(note: converse is false!)

Proof.

let P be a shortest u-v path in G. let s, t be two intermed. vertices on P



let P' be sub-seq. from s to t.
must show P' is a shortest s-t path.

Assume, to get a $\dot{\times}$, that P' is not a shortest s-t Path. Then there is a shorter one, call it P''

i.e. $\text{len}(P'') < \text{len}(P')$.

Then consider sequence

$$P_{us} + P''_{st} + P_{tv} \ldots \ldots$$

(may not be a Path, but is a Walk.)

We have

$$len(P_{us} + P''_{st} + P_{tv})$$

$$= len(P_{us}) + len(P''_{st}) + len(P_{tv})$$

$$< len(P_{us}) + len(P'_{st}) + len(P_{tv})$$

$$= len(P_{us} + P'_{st} + P_{tv})$$

$$= len(P),$$

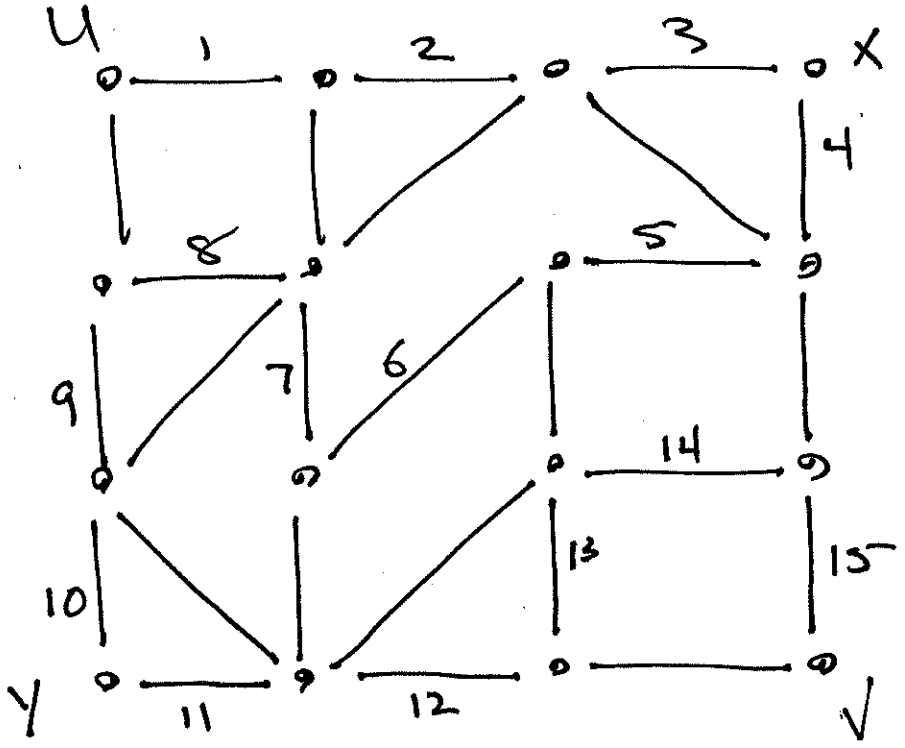
contradicting that P is a shortest u-v path. So this $\not\exists$ shows no such path P'' can exist. $\therefore P'$ is a shortest s-t path. ∎

so Problem 1 "exhibits optimal sub-structure"!

Problem 2 does not!

Ex.

$d(u,v) = 5$



$l(u,v) = 15$ but $l(x,y) \geq 14$

so sub-seq. of a longest Path is not necessarily a longest Path.

Procedure for designing a Dyn. Programming algorithm.

1. Characterize the optimal substructure, i.e. show that solving entails making a seq. of choices, each leaving a sub-instance.
2. Recursively define value of optimal soln in terms of optimal sub-instance solutions (write a recursive formula)
3. Compute value of an optimal soln in a bottom-up fashion i.e. fill in a table.

4. Construct an optimal solution by backtracking through the table.

Problem Matrix Chain multiplication

$$A_1 A_2 A_3 \cdots A_n$$

start with : $A \cdot B$

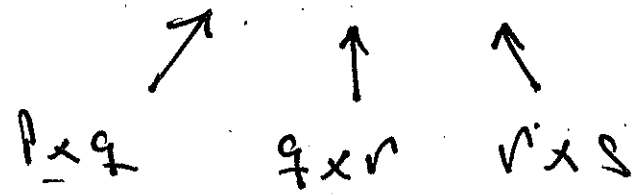
for $\begin{pmatrix} 1 \le i \le p \\ 1 \le j \le r \end{pmatrix}$ $\quad p \times q \quad q \times r$

$$(A \cdot B)_{ij} = \sum_{k=1}^{q} a_{ik} \cdot b_{kj} \quad \begin{pmatrix} q \text{ basic} \\ \text{OPS.} \end{pmatrix}$$

basic operation : real multiplication

total cost $= pqr$

What is cost of $A \cdot B \cdot C$



(1) $A(BC)$: $pqs + qrs$

or

(2) $(AB)C$: $pqr + prs$

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

Then

(1) has cost 200,000

(2) " " 20,000

Exercise

write the 5 different Parenthesizations of: ABCD

Exercise

Let $P(n)$ = # of Parenthesizations of $A_1 A_2 \cdots A_n$. Show that $P(n)$ satisfies

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

Hard exercise: show

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

note: $C_n = \frac{1}{n+1}\binom{2n}{n}$ is called Catalan Numbers.

note: $\binom{2n}{n} = \Theta\left(\frac{4^n}{n^{3/2}}\right)$

Problem:
Given $n \geq 1$ and a matrix-chain product $A_1 A_2 \cdots A_n$, where

A_i has size $p_{i-1} \times p_i$ $(1 \leq i \leq n)$

determine a parenthesization that minimizes total # of basic ops.

Actually 2 problems

- Value of opt. soln, i.e. min Poss. # of real multiplications
- Opt. soln itself: a Parenthesization that achieves min.

Typical Subinstance

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

where $1 \leq i \leq j \leq n$

Note: any opt. Paren. of $A_i \cdot \cdots \cdot A_j$ has a first split Point k:

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

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

claim:

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

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

Proof.

Assume, to get a $\forall$, that $(A_i \cdots A_k)$ is not optimally parenthesized. Then there is a better paren. than the one inherited from the opt. paren. of $A_i \cdots A_j$. Substitute that better paren. of $(A_i \cdots A_k)$ to get a better paren. of $A_i \cdots A_j$) which is impossible. ∎

Exercise show converse is false, i.e. Concatenation of Opt. Parens. is not necessarily optimal.

Let $m[i,j]$ = min # of real multiplications needed to compute $A_i \cdots A_j$

where $1 \le i \le j \le n$. Our Goal is to find $m[1,n]$.

note: if $i = j$, then initialize

$$m[i,i] = 0$$

By Proof ...

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