CSE 102 5-14-24

- mid 2. Thur. May 23
- Project. Mon. May ~~20~~ 27

<u>Principle of Optimality</u>

or <u>Optimal Substructure</u>.

an optimal solution to any instance is a combination of optimal sub-instance solutions.

- Coin change

$$C[i,j] = \min\left(C[i-1,j],\ 1+C[i,j-d_i]\right)$$

- Discrete Knapsack

$$V[i,j] = \max\left(V[i-1,j],\ v_i + V[i-1,j-w_i]\right)$$

Graph Theory:

Problem 1:

Given a graph G and $u,v \in V(G)$, find a shortest u-v path.

- what is length: $d(u,v)$
- what is the shortest path.

Problem 2

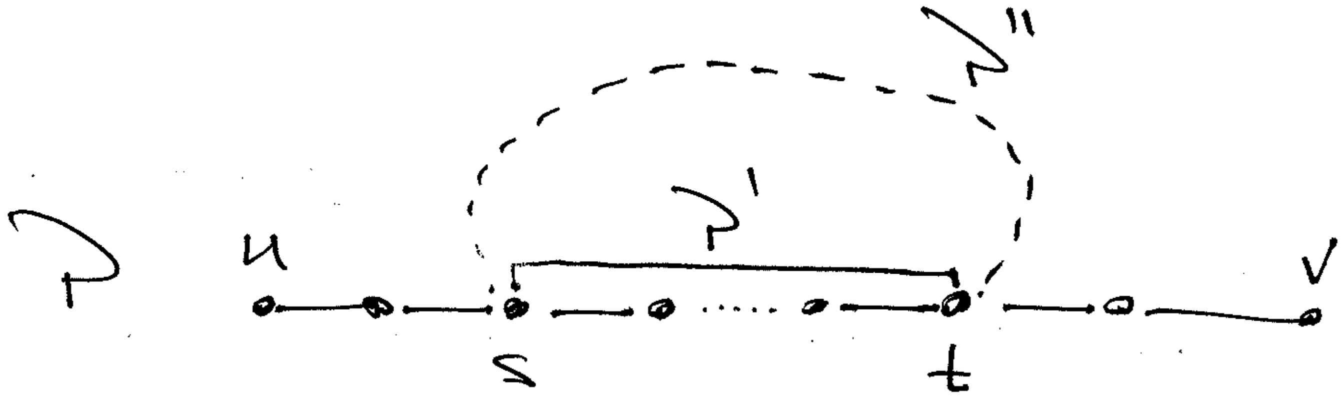
Given a graph G and $u, v \in V(G)$, find a longest u-v Path.

- what is len.: $l(u,v)$
- " " the longest Path.

Lemma

Problem 1 exhibits optimal Substructure.
i.e. any subPath of a shortest Path is a shortest Path.

Proof.
Let P be a shortest u-v Path in G.



Let s and t be any 2 intermediate vertices. Let P' be the sub-path from s to t. Assume P' is not a shortest s-t path in G. Then there exists a shorter s-t path, call it P''.

Let Q be the u-v path

$$Q : u \xrightarrow[P]{} s \xrightarrow[P'']{} t \xrightarrow[P]{} v$$

Then $\text{len}(P) - \text{len}(Q) = \text{len}(P') - \text{len}(P'') > 0$

$\therefore$ $\text{len}(Q) < \text{len}(P)$, a $\times$ ■

note: Problem 2 does not exhibit optimal substructure.

Procedure for Dynamic Programming Solutions.

1. characterize the structure of an opt. solution as consisting of a combination of opt. sub-instance solutions.

2. recursively define optimal solutions in terms of sub-instance solutions.

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

4. Construct an optimal solution by backtracking through the table in (3).

Ex. Matrix Chain Multiplication.

$$\underbrace{(A_1 A_2 \cdots A_i)}_{\text{Subinstance}}\underbrace{(A_{i+1} \cdots A_{n-1} A_n)}_{\text{subinstance}}$$

split point: i