Lec 9/21

Thursday, September 21, 2017 12:47

Non-recursive DP: do base case first.

Method 3:

$(x_1, \dots, x_{2n})$ is a seq. of decisions:

take a_i, b_j if $a_i = b_j$.

remove one (which one).

$$L(i,j) = \begin{cases} 1 + L(i+1, j+1) & \text{if } a_i = b_j \\ \max\{L(i+1,j),\ L(i,j+1)\} & \text{if } a_i \neq b_j \end{cases}$$

$L(n+1, j) = 0 = L(i, n+1)$

runtime $\Theta(n^2)$.

Forward/Backward both work for all versions.

All-pair Shortest paths:

for every pair of nodes $(u,v) \in V^2$ find the

shortest path from u to v in E in $G(V,E)$ (weighted, directed).

$\forall u \in V$, construct $(x_1, \dots, x_k)$

Approach 1:

X_1 : the next node:

$\deg u$ = # choices

$$L(i,j) = \min\{d(i,z) + L(z,j) : (i,z) \in E\}$$

Doesn't work bc cycles.

$L^k(i,j)$ = length of a shortest path from i to j which has at most k intermediate nodes.

$$L^k(i,j) = \min\{d(i,z) + L^{k-1}(z,j) : (i,z) \in E)$$

$$L^0(i,j) = \begin{cases} 0 & \text{if } i=j \\ d(i,j) & \text{if } (i,j) \in E \\ \infty & \text{if } (i,j) \notin E \end{cases}$$

takes $\Theta(n^4)$ time. the more parameters in f the more time it takes.

Approach 2:

X_1: going thru node 1 (enter & leave) or not?

$D^k(i,j)$ = length of shortest path from i to j w/ intermediate nodes $\in \{1, \dots, k\}$.

$$D^k(i,j) = \min\{D^{k-1}(i,j),\ D^{k-1}(i,k) + D^{k-1}(k,j)\}$$

$$D^0(i,j) = \begin{cases} \text{weight of } (i,j) & \text{if } (i,j) \in E \\ 0 & \text{if } i=j \\ \infty & \text{otherwise} \end{cases}$$

$\Theta(n^3)$

goal: $D^n(i,j)$ gives shortest distance from i to j.

Problem 3 on HW:

Path(k, i, j) should have no loops.