RAHUL ETHIRAJ
ethiraj@usc.edu

CSCI-570- Spring 2018 - HW7

1.)

$$OPT(v) = \max_{u:v \in adj[u]} \{OPT(u) + C(u,v)\}$$

v in topological order

Base cases:

$OPT(s) = 0$
$OPT(v) = -\infty$

Solution: Find largest of $OPT(v)$, backtrack and find the largest path.

$\Rightarrow$ Complexity: $O(m+n)$

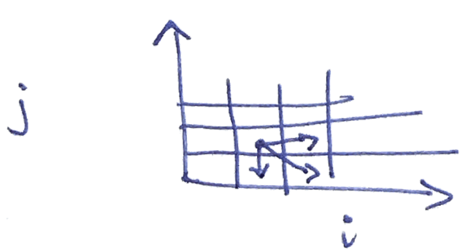
2.)

Sol.1: Find LCS of S & S^R

$LCS[i,j] = 1 + LCS[i-1, j-1]$, if $S[i] = T[j]$
$LCS[i,j] =$ max $(LCS[i-1, j], LCS[i, j-1])$, if $S[i] \neq T[j]$

Sol 2: :

$$OPT(i,j) = \begin{cases} \text{if } S_i = S_j, & OPT(i+1, j-1) + 2 \\ \text{else}, & \max(OPT(i+1, j), OPT(i, j-1)) \end{cases}$$



Move to the cell according to the opt directions, to set the ~~ans~~ answer, through backtracking.

$\Rightarrow$ Complexity = $\Theta(n^2)$

SRUTHI ♥ RAHUL

3.) Base cases :

maxCoin [i] [i] = coins[i]

left = if (i+2 <= j)
maxCoin [i+2] [j]
else
0

bottom = if (j-2 >= i)
maxCoin [i] [j-2]
else
0

maxCoin[i][j] = max (coins[i] + min (left , maxCoin [i+1][j-1])
, (coins [j] + min (maxCoin [i+1] [j-1], bottom)

$\Rightarrow$ Complexity : $\Theta(n^2)$

4)

$0 \leq i \leq n-1$

cutRod (n) = max (price[i] + cutRod (n-i-1))

$\Rightarrow$ Complexity = $\Theta(n)$