HW 1 Qiang Shi ID# 01664540

Problem 1

MEMOIZED-CUT-ROD (p,n)

Let r[0..n] be a new array

for i = 1 to n

r[i] = - ∞

return MEMOIZED-CUT-ROD-AUX (p,n,r)

MEMOIZED-CUT-ROD-AUX (p,n,r)

if r[n] ≥ 0

return r[n]

if n== 0

q = 0

else q = - ∞

for i = 1 to n-1

q = max (q,p[i]+ MEMOIZED-CUT-ROD-AUX (p,n-1,r)-c)

q = max (q,p[n])

r[n] =q

return q

------------------------------------------------------------------------------------------------

BOTTOMUP-CUT-ROD (p,n)

Let r[0..n] be a new array

r[0] = 0

for j = 1 to n

q = - ∞

for i = 1 to j

if j==i

q = max (q, p[i] + r[j-i])

else

q = max (q, p[i] + r[j-i] - c)

r[j] =q

return r[n]

----------------------------------------------------------------------------------------------------------

problem 2

LCS\_AUX(counter, memo, i, j, X, Y):  
 **if** memo[i,j]>=0:  
 v = memo[i,j]  
 **else if** i==0 **or** j==0:  
 v = 0  
 **else if** X[i] == Y[j]:  
 counter = counter + 4  
 v = LCS\_AUX(counter, memo, i - 1, j - 1, X, Y) + 1  
 **else**:  
 counter = counter -1  
 v = max(LCS\_AUX(counter, memo, i - 1, j, X, Y), LCS\_AUX(counter, memo, i, j - 1,

X, Y))  
 memo[i,j]= v  
 **return** v

----------------------------------------------------------------------------------

Problem 3

Yes, it has.

MATRIX-CHAIN-ORDER.(p)

n = p.length – 1

let m[1..n,1..n] and s [1..n-1,2..n] be new tables

for i = 1 to n

m[i,i] = 0

for l = 2 to n

for i = 1 to n-l+1

j = i + l -1

m[i,j] = 0

for k = i to j -1

q = m[i,k]+m[k+1,j]+

if q>m[i,j]

m[i,j]=q

s[i,j] = k

return m and s

problem 4

LPS(i,j) = 2 +LPS(i+1,j-1), if S[i] = S[j]

Max{ LPS(i+1,j-1), LPS(i+1,j-1)} if S[i] ≠ S[j]

LPS(X)

n = X.Length

let c[1..n,1..n] new tables

for i = n-1 down to 0

if X[i] = X[j]

f[i,j] = f[i+1][j-1]+2

else

f[i,j] = max(f[i][j-1],f[i+1][j])

the cost is O(n^2)

c[i+1,j-1] +2 if i,j>0 &xi = yj

LPS(i,j,X[1..n])

Counter = 0

MaxCounter = 0

If j – i >1

If X[i] ==X[j]

Counter ++

If MaxCounter < Counter

MaxCounter =Counter