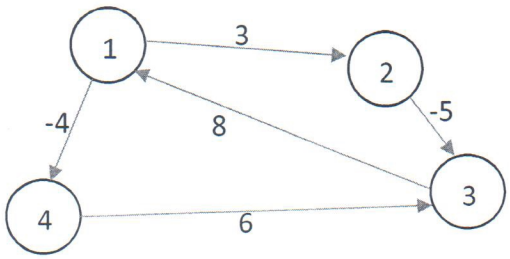
CSCE 500 Final Exam

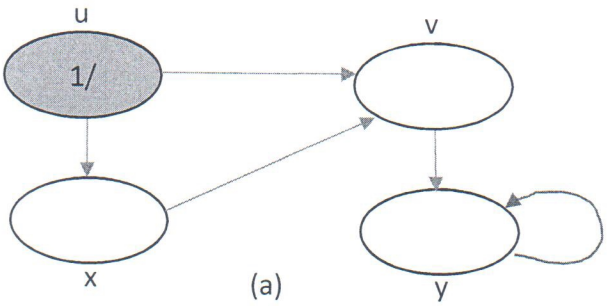
1. All-pairs paths (APSP) can be derived by extending the number of links per path repeatedly. A fast version for APSP doubles the number of links in each iteration.
   a. Derive the resulting distance matrix of the directed graph below, following the fast APSP (Show the intermediate matrix of each iteration.)



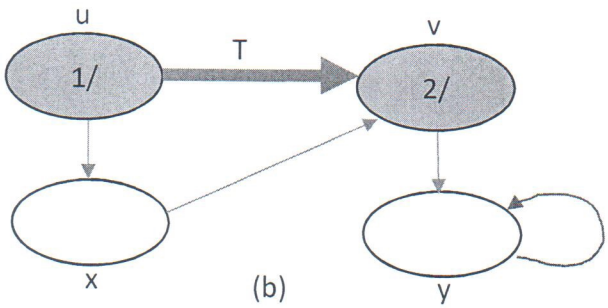
$$L^{(1)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -5 & \infty \\ 8 & \infty & 0 & \infty \\ \infty & \infty & 6 & 0 \end{pmatrix} \quad L^{(2)} = \begin{pmatrix} 0 & 3 & -2 & -4 \\ 3 & 0 & -5 & \infty \\ 8 & 11 & 0 & 4 \\ 14 & \infty & 6 & 0 \end{pmatrix} \quad L^{(4)} = \begin{pmatrix} 0 & 3 & -2 & -4 \\ 3 & 0 & -5 & -1 \\ 8 & 11 & 0 & 4 \\ 14 & 17 & 6 & 0 \end{pmatrix}$$

   b. How many iterations does it take to obtain the resulting matrix for a graph with n vertexes. The number of iterations is $\lceil \lg(n-1) \rceil$.

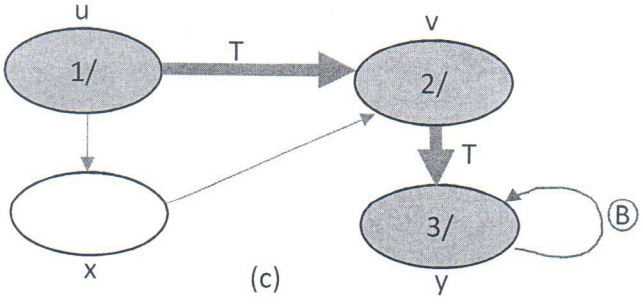
2. Depth First Search (DFS) colors every vertex of a given graph in white, gray, and black during the process to build a search tree, with a discovery time and a finish time kept at each vertex. All edges in a directed graph G = (V, E) under DFS are classified into four types: tree, back, forward and cross edges. For G given below, conduct DFS that starts from vertex u at time clock = 1 to
   i. mark the discovery time and the finish time of every vertex, and
   ii. classify the types of all edges. (Note: one intermediate result should be shown upon each edge classification for clarity.)



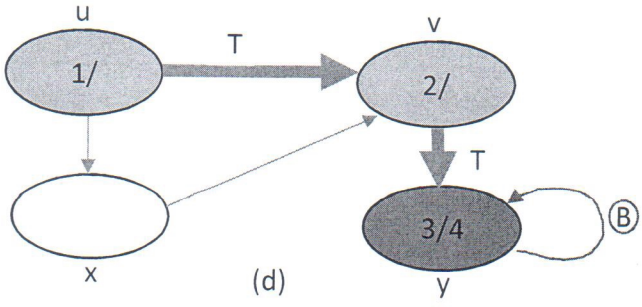
(a)



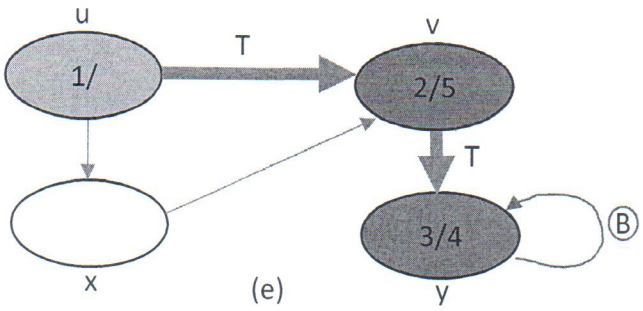
(b)

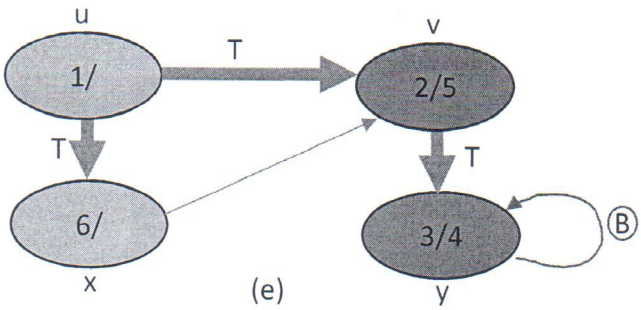


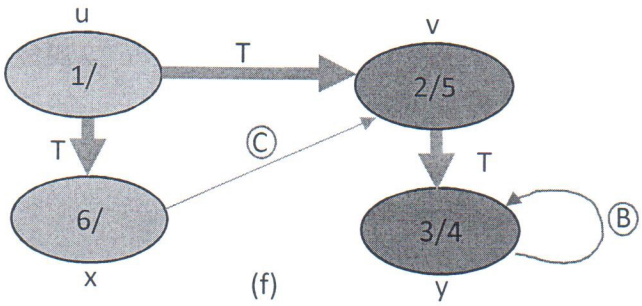
(c)

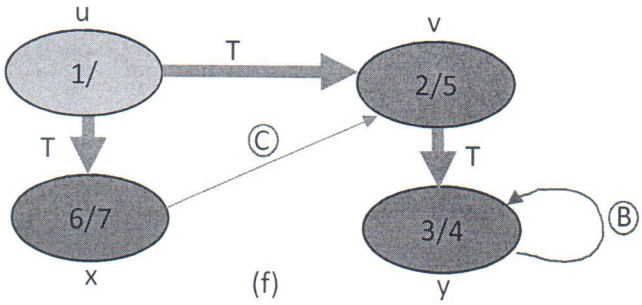


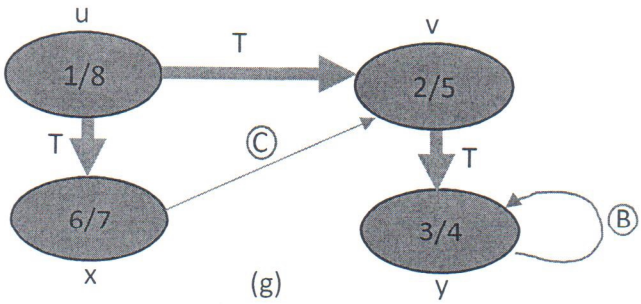
(d)


u
v
1/
T
2/5
T
3/4
B
x
y
(e)


u
v
1/
T
2/5
T
T
6/
3/4
B
x
y
(e)

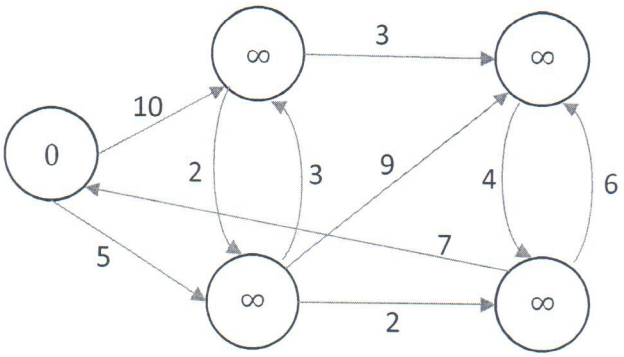

u
v
1/
T
2/5
T
C
T
6/
3/4
B
x
y
(f)


u
v
1/
T
2/5
T
C
T
6/7
3/4
B
x
y
(f)

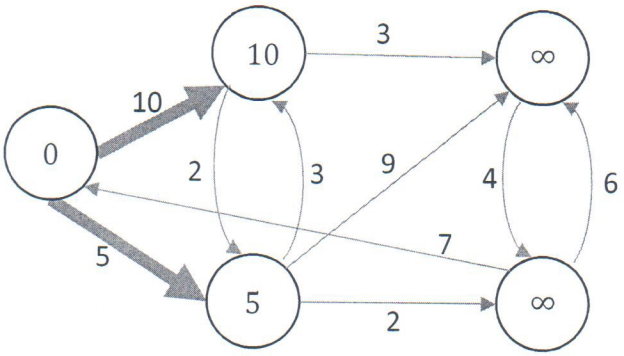

u
v
1/8
T
2/5
T
C
T
6/7
3/4
B
x
y
(g)

3. The Bellman-Ford algorithm (BF) solves the single-source shortest path problem in a weighted directed graph G = (V,E). Given the graph G below, follow BF to find shortest paths from vertex s to all other vertexes, with all predecessor edges shaded an estimated distance values from s to all vertexes provided at the end of each iteration.

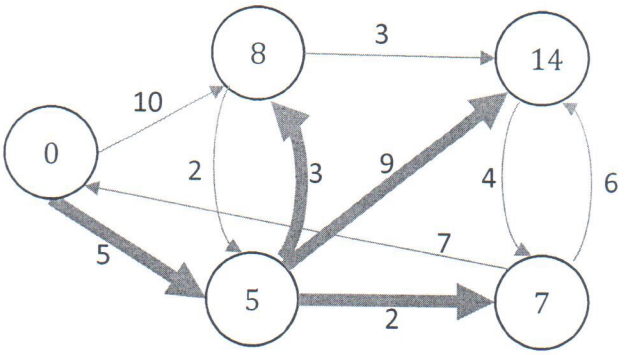
   How iterations are involved in BF for a general graph G = (V,E)? |G.V.| - 1



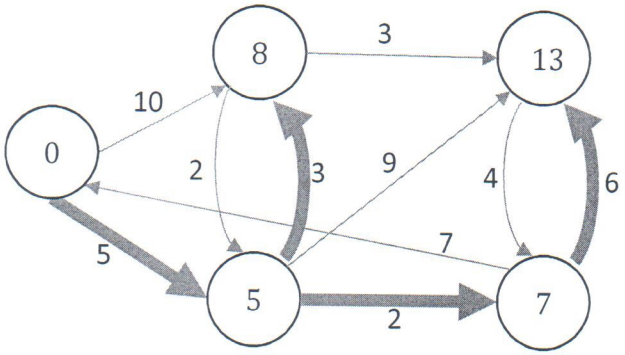
(a)



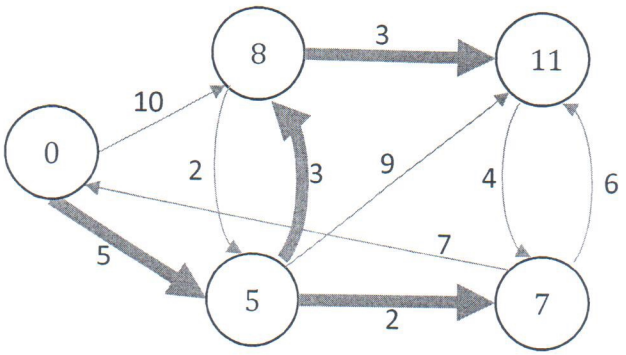
(b)



(c)



(d)

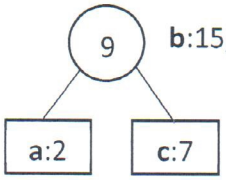


(e)

4. How your construction of an optimal Huffman code for the set of 6 frequencies: **a**:2, **b**:15, **c**:7, **d**:21, **e**:17, **f**:38

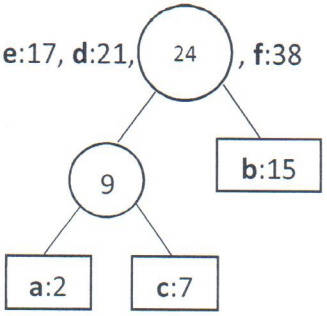
   Sort in non-decreasing frequencies: **a**:2, **c**:7, **b**:15, **e**:17, **d**:21, **f**:38

(a)



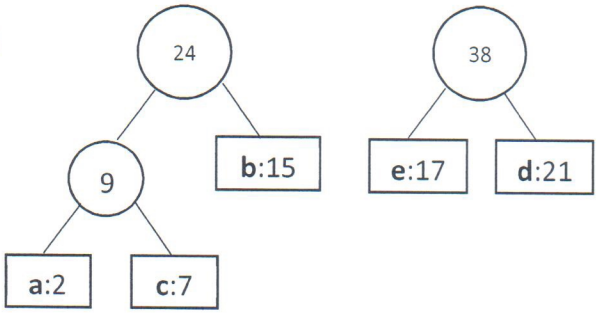
b:15, **e**:17, **d**:21, **f**:38

(b) **e**:17, **d**:21,



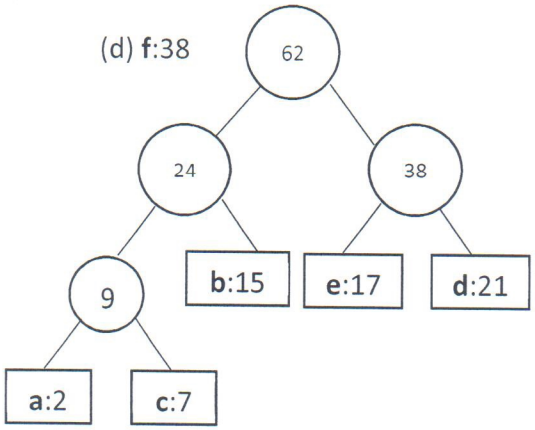
, **f**:38

(c)

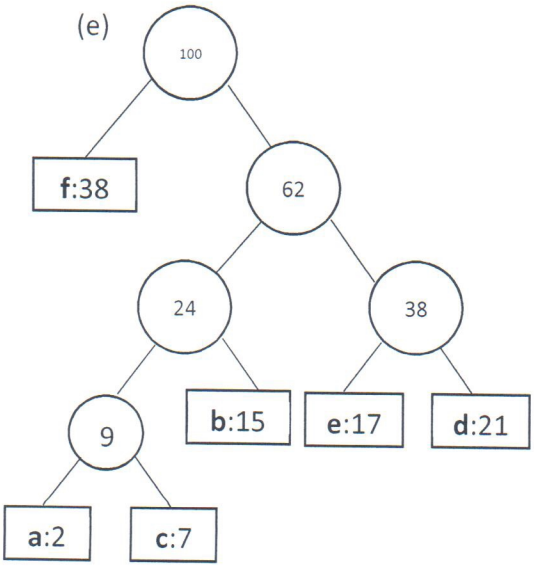


, **f**:38

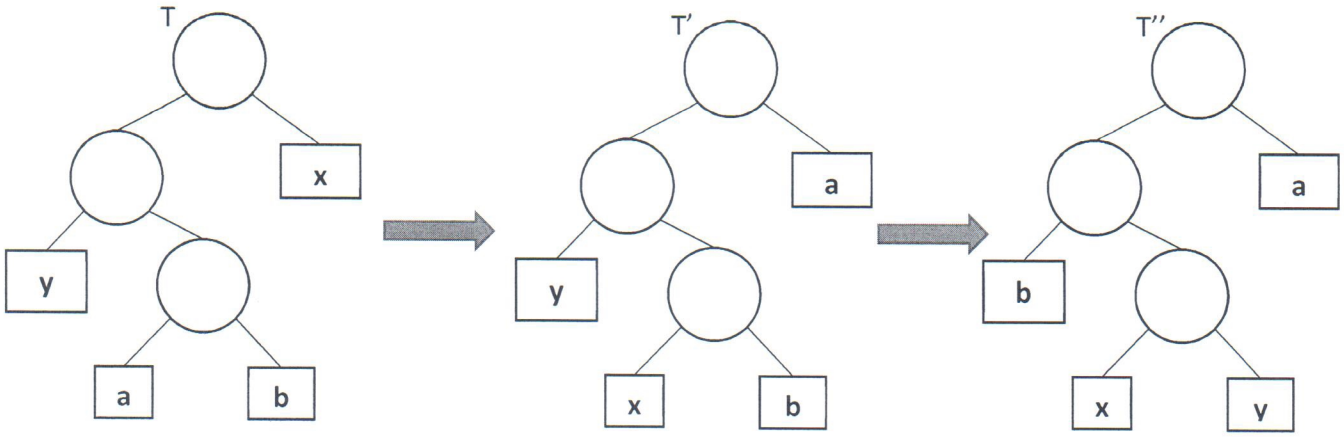
(d) **f**:38



(e)



In general, for an alphabet set C, where each of tis element c has frequency c.f, briefly prove that there exists an optimal prefix code for C such that the codewords of its two elements x and y have the same length and differ only in the last bit. (Hint: transform an arbitrary optimal prefix code tree where x and y are not under the same parent node to another tree in which they are.)



T denotes an arbitrary optimal prefix code.

$$B(T) - B(T') = \sum_{c \in C} c.freq \cdot d_T(c) - \sum_{c \in C} c.freq \cdot d_{T'}(c)$$

$$B(T) - B(T') = x.freq \cdot d_T(x) + a.freq \cdot d_T(a) - x.freq \cdot d_{T'}(x) + a.freq \cdot d_{T'}(a)$$

$$B(T) - B(T') = x.freq \cdot d_T(x) + a.freq \cdot d_T(a) - x.freq \cdot d_T(a) + a.freq \cdot d_T(x)$$

$$B(T) - B(T') = (a.freq - x.freq)\big(d_T(a) - d_T(x)\big)$$

$$B(T) - B(T') \geq 0, therefore, B(T) \geq B(T')$$

$$But, B(T') \geq B(T), since\ T\ is\ optimal, so\ we\ have\ B(T') \equiv B(T)$$

Similarly, $B(T') \geq B(T'')$ *to yield* $B(T) \geq B(T'), since\ B(T') \equiv B(T).$

Given that T is optimal, we have B(T'') ≡ B(T), implying that T"" is another optimal prefix code with x and y differing only in the last bit.

5. Solve the recurrence of $T(n) = 3 \cdot T\left(\frac{n}{4}\right) + n^{1/2}$

   Using the Master Theorem: a =3, b =4. Since $n^{1/2} = O\left(n^{\log_4 3}\right)$, $T(n) = n^{\log_4 3}$

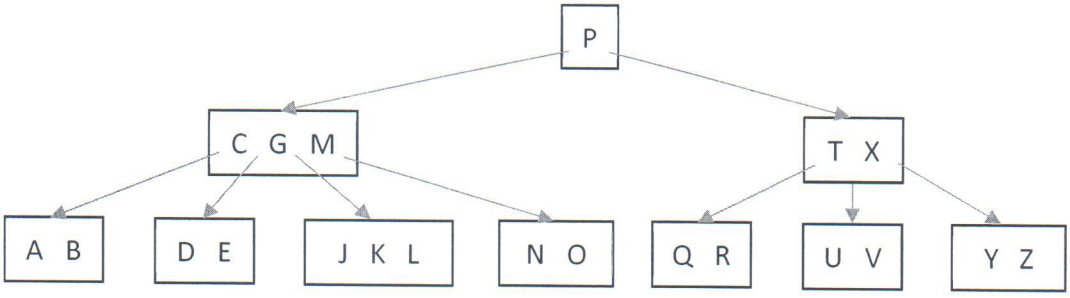
6. The Given two hash functions h_1 and h_2 for Cuckoo hashing under two tables, T_1 and T_2, describe the steps involved in inserting a record with the key of K_{new}.

   Cuckoo hashing can be analyzed by the Cuckoo graph, whose nodes denote table entries and links connect pairs of nodes where given keys can be held. State when a new key can be inserted successfully based on the Cuckoo graph.

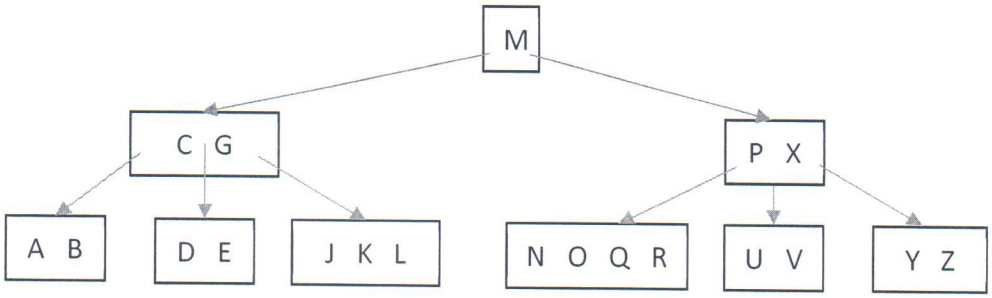
   Hash function h_1 is applied to key K_{new}. The result is where this key is stored in table T_1. If a key is already store in that location, key Knew is store there and the hash function h_2 is applied to the key that was stored there and the result is used to store the displaced key into table T_2. If a key is already store in that location, that key is store there and the hash function h1 is applied to the key that was stored there and the result is used to store the displaced key into table T_1. In the event that a key is already stored in that table T_1 location, the process continues until an empty key location is found, or the key cannot be inserted.

   Insertion succeeds if and only if the new edge (defined by the new key) contains at most one cycle.

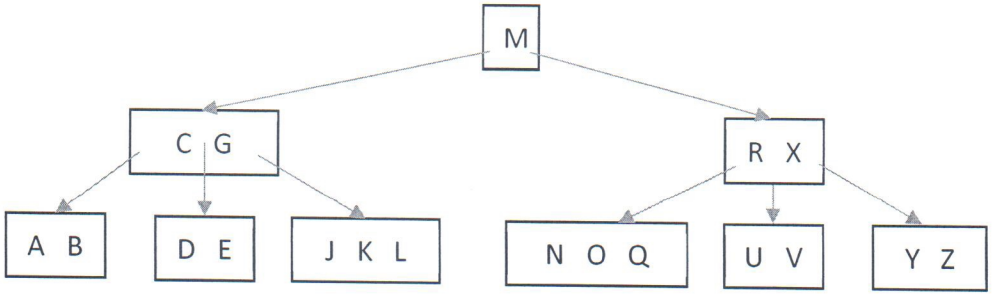
7. Given a B-tree with the minimum degree of t=3 below, show the results after
   a. deleting T and the deleting P in order,
   b. followed by inserting S.



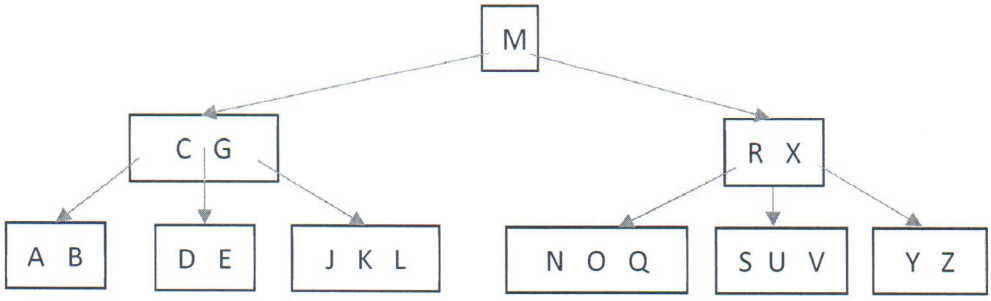
Deleting T...



Deleting P...



Inserting S...



8. Consider the matrix-chain multiplication problem for four matrices A1, A2, A3, A4, with their sizes being 30x10, 10x20, 20x50, and 50x40, respectively. Follow the tabular, bottom-up method in the procedure for MATRIX-CHAIN-ORDER below to construct tables that keeps respectively entry m[i, j] for all $1 \le i, j \le 4$ and entry s[i,j] for $1 \le i \le 3$ and $2 \le j \le 4$ to get the optimal parenthesized multiplication result.
   a. Construct the two tables, with their entry values shown.
   b. Show the parenthesized multiplication of the matrix-chain.

MATRIX-CHAIN-ORDER(p)

```
1.  n = p.length -1
2.  let m[1..n, 1..n] and s[1..n-1, 2..n] be new tables
3.  for i = 1 to n
4.      m[i, i] = 0
5.  for ℓ = 2 to n                    // l is the chain length
6.      for i = 1 to n – ℓ + 1
7.          j = i + ℓ - 1
8.          m[i, j] = ∞
9.          for k = i to j – 1
10.         q = m[i,k] + m[k+1,j] + p_{i-1} p_k p_j
11.         if q < m[i,j]
12.             m[i,j] = q
13.             s[i,j] = k
14. return m and s
```

m

j \ i	1	2	3	4
4	42,000	30,000	40,000	0
3	20,000	10,000	0	
2	6,000	0		
1	0			

$m[1,2] = m[1,1] + m[2,2] + p_0p_1p_2 = 0 + 0 + 30x10x20 = 6,000$
$m[2,3] = m[2,2] + m[3,3] + p_1p_2p_3 = 0 + 0 + 10x20x50 = 10,000$
$m[3,4] = m[3,3] + m[4,4] + p_2p_3p_4 = 0 + 0 + 20x50x40 = 400,000$
$m[1,3] = \min\{m[1,1] + m[2,3] + p_0p_1p_3,\ m[1,2] + m[3,3] + p_0p_2p_3\} = \min\{20,000;36,000\}=20,000,\ k=1$
$m[2,4] = \min\{m[2,2] + m[3,4] + p_1p_2p_4,\ m[2,3] + m[4,4] + p_1p_3p_4\} = \min\{48,000;30,000\}=30,000,\ k=3$
$m[1,4] = \min\{m[1,1] + m[2,4] + p_0p_1p_4,\ m[1,2] + m[3,4] + p_0p_2p_4,\ m[1,3] + m[4,4] + p_0p_3p_4\}$
$m[1,4] = \min\{42,000;\ 64,000;\ 80,000\} = 42,000,\ k=1$

$A_1 \cdot ((A_2 \cdot A_3) \cdot A_4)$

s i

j \ i	1	2	3
4	1	3	3
3	1	2	
2	1		