CSCE 500 Midterm Exam #2

11/14/2018

1. The recurrence of Procedure CUT-ROD(p,n) is given by $T(n) = 1 + \sum_{j=0}^{n-1} T(j)$, with T(0) = 1. Solve T(n) (12%).

T(0) = 1

T(1) = 1 + T(0)
= 2

T(2) = 1 + T(0) + T(1)
_,____/
= T(1) + T(1)
= 2*T(1)
= 4

T(3) = 1 + T(0) + T(1) + T(2)
_,_____/
= T(2) + T(2)
= 2*T(2)
= 8

T(4) = 1 + T(0) + T(1) + T(2) + T(3)
_,_____/
= T(3) + T(3)
= 2*T(3)
= 16

:

T(n) = 2*T(n-1) = 2^n

2. The problem of optimal parenthesization over a chain of matric multiplications can be solved by a divide-and-conquer approach recursively. Let $m[i,j]$ denote the minimum number of scalar multiplications needed to compute $A_i \bullet A_{i+1} \bullet A_{i+2} \bullet \ldots A_j$, with A_k sized as $p_{k-1} \times p_k$ (for $i \leq k \leq j$), give the recurrence definite of the problem. (10%)

$$\boldsymbol{m[i,j] = m[i,k] + m[k+1,j] + p_{i-1}p_k p_j \text{ , if } i < j,}$$

$$\boldsymbol{m[i,j] = 0 \text{ if } i = j}$$

The recurrence leads to exponential complexity but can be solved by dynamic programming much faster. What is the resulting time complexity and how do you get that complexity result? (6%)

There are $\theta(n^2)$ distinct subproblems, to yield a time complexity of $\theta(n^3)$. This is because it has 3 nested loops of $\theta(n)$.

3. Given a set of 4 keys, with the following probabilities, determine the cost and structure of an optimal binary search tree, following the tabular, bottom-up method realized in the procedure of OPTIMAL-BST below to construct and fill e[1..5, 0..4], w[1..5, 0..4] and root[1..4, 1..4].

i	0	1	2	3	4
p_i		0.12	0.08	0.15	0.18
q_i	0.08	0.06	0.09	0.10	0.14

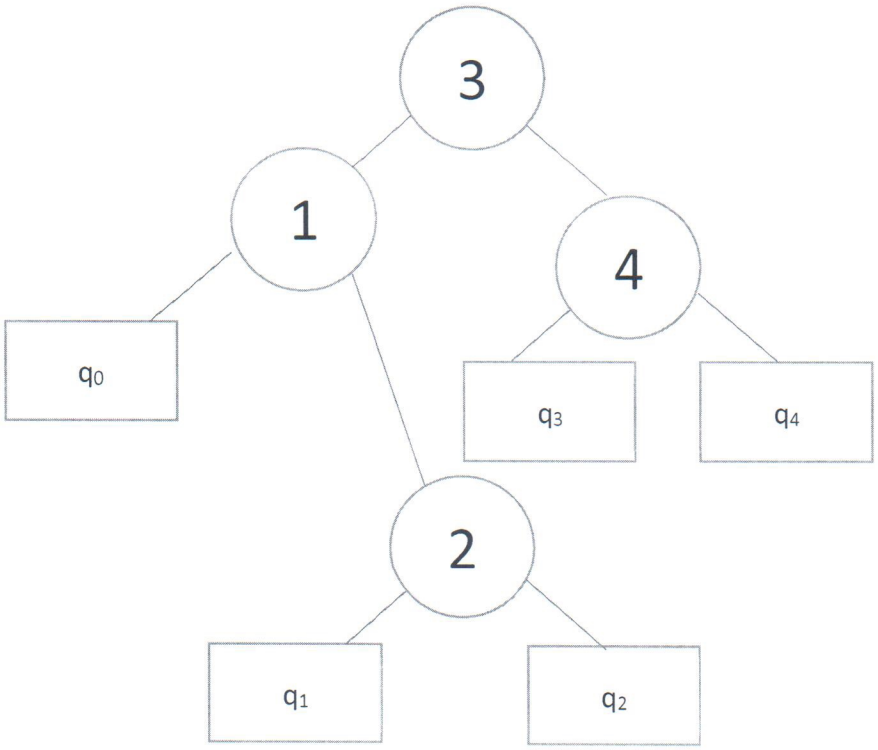
a. Fill in the missing 3 statements in the procedure. (6%)
   1. $t = e[i,r-1] + e[r+1,j] + w[i,j]$
   2. $e[i,j] = t$
   3. $root[i,j] = r$
b. Construct and fill the three tables and show the optimal BST obtained. (30%)

e	i					
		1	2	3	4	5
j	4	2.55	1.84	1.33	0.66	0.14
	3	1.61	0.96	0.53	0.10	
	2	1.89	0.38	0.09		
	1	0.4	0.06			
	0	0.08				

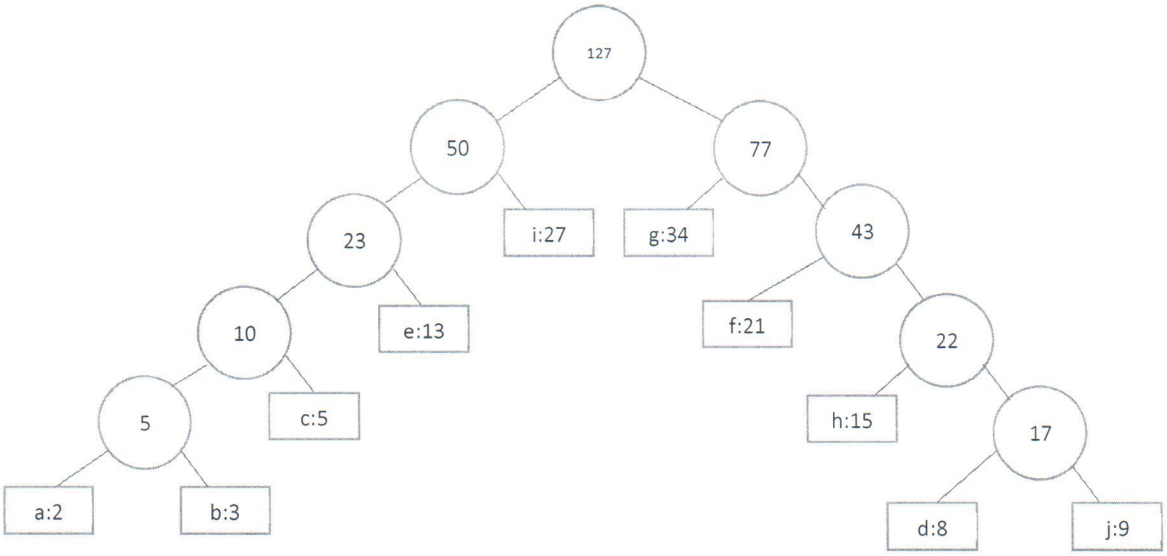
root	i					
		1	2	3	4	
j	4	3	3	4	4	
	3	2	3	3		
	2	1	2			
	1	1				

w	i					
j		1	2	3	4	5
	4	1.0	0.80	0.66	0.42	0.14
	3	0.68	0.48	0.34	0.10	
	2	0.43	0.23	0.09		
	1	0.26	0.06			
	0	0.08				

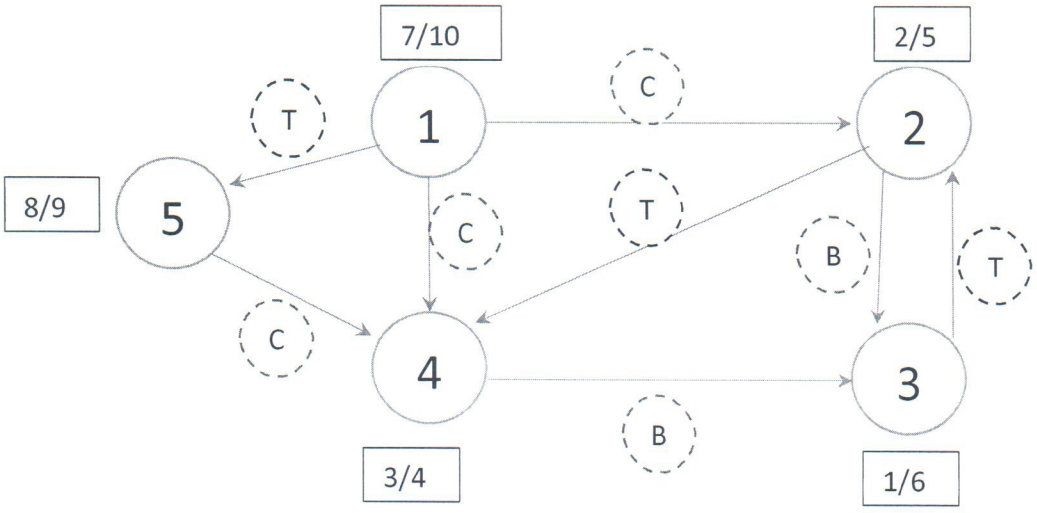
OBT Structure



4. Show your construction of an optimal Huffman code for the set of 7 frequencies: **a**:2, **b**:3, **c**:5, **d**:8, **e**:13, **f**:21, **g**:34, **h**:15, **i**:27, j:9 (10%)



5. Follow depth-first search (DFS), starting from Node 3, to traverse the graph shown below. Mark (1) the type of every edge and (2) the discovery and finish times of each node. (10%)



6. The Dijkstra's algorithm (DIJ) solves the single-source shortest-path problem in a weighted directed graph G=(V,E). Given the graph G below, follow DIJ to find the shortest paths from vertex s to all other vertexes, with predecessor edges shaded and estimated distance values from s to all vertexes provided at the end of each iteration. (12%)

   Minimum distance from source node 0

   s – 0
   t – 8
   x – 9
   y –5
   z – 7

   Edges in the predecessor tree: (s,y), (y,t), (y,z), (y,x) and (t,x)

   What is the time complexity of DIJ for a general graph G=(V,E), if candidate vertexes are kept in a binary min-heap? (4%)

   The time complexity of the DIJ if the candidate vertexes are kept in a binary min-heap is O((V+E)lg V).