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%)

2. The problem of optimal parenthesization over a chain of matrix 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 $A_i \bullet A_{i+1} \bullet A_{i+2} \bullet \cdots \bullet A_j$, with A_k sized as $p_{k-1} \times p_k$ (for $i \le k \le j$), give the recurrence definition of the problem. (10%)

   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%)

3. Given a set of 4 keys, with the following probabilities, determine the cost and the 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%)

(b) Construct and fill the three tables, and show the optimal BST obtained. (30%)

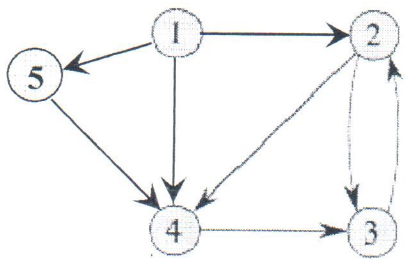
```
OPTIMAL-BST(p, q, n)
let e[1..n+1, 0..n], w[1..n+1, 0..n],
        and root[1..n, 1..n] be new tables
for i = 1 to n+1
    e[i, i-1] = q_{i-1}
    w[i, i-1] = q_{i-1}
for l = 1 to n
    for i = 1 to n-l+1
        j = i+l-1
        e[i, j] = ∞
        w[i, j] = w[i, j-1] + p_j + q_j
        for r = i to j
            1.
            if t < e[i, j]
                2.
                3.
return e and root
```

44

1 2

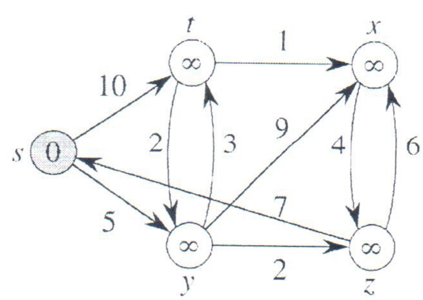
4. Show your construction of an optimal Huffman code for the set of 7 frequencies: **a**:2 **b**:3 **c**:5 **d**:8 **e**:13 **f**:21 **g**:34 **h**:15 **i**:27 **j**: 9. (10%)

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 the 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 shortest paths from vertex *s* to all other vertexes, with all predecessor edges shaded and estimated distance values from *s* to all vertexes provided at the end of each iteration. (12%)

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%)



Good luck!