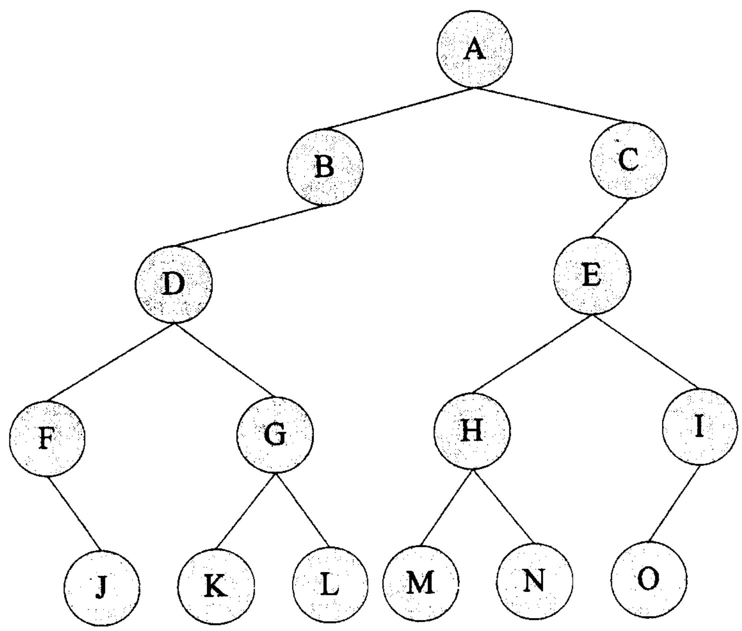
編號： 381　系所：資訊管理研究所乙組　　科目：資料結構

本試題是否可以使用計算機： □可使用 ， ☑不可使用 （請命題老師勾選）

1. When keys can be equal, each comparison may have three results instead of two: $K_i < K_j$, $K_i = K_j$, $K_i > K_j$. Sorting algorithms for this general situation can be represented as extended ternary trees, in which each internal node i:j has three subtrees; the left, middle, and right subtrees correspond respectively to the three possible outcomes of the comparison.
Draw an extended ternary tree which defines a sorting algorithm for n=3, when equal keys are allowed. There should be 13 external nodes. (15%)

2. Given the following input data in order:
503, 087, 512, 061, 908, 170, 897, 275, 653
Please construct:
(a) AVL tree (5%)
(b) 2-3 tree (5%)
(c) Deap (5%)

3. For binary tree traversal, show the list of letters by inorder and postorder. Can we uniquely determine a binary tree by preorder or postorder? Explain your answer. (10%)



4. Describe the concept of Quick Sort and analyze the best and the worst case time complexities. (10%)

（背面仍有題目,請繼續作答）

5. (24%) **True** or **False**, and **EXPLAIN**
Circle T or F for each of the following statements to indicate whether the statement is true or false, respectively. If the statement is correct, briefly state why. If the statement is wrong, explain why or give a counter example. Answers without reasons will get at most 1 point.

(a) (**T** , **F**) A binary search tree of n nodes will have an $O(n \lg n)$ height.
(b) (**T** , **F**) To select the ith smallest number among n distinct random numbers requires $\Omega(n \lg n)$ time.
(c) (**T** , **F**) Sorting n distinct integers in the range $[0, n^4 - 1]$ by counting sort can be done in $O(n)$ time.
(d) (**T** , **F**) In a red-black tree, let α and β be the length for its shortest and longest path, respectively. Then $2\alpha \geq \beta$.
(e) (**T** , **F**) There exists a data structure to maintain a dynamic set S with operations INSERT(x, S), DELETE(x, S), and MEMBER(x, S) that has an expected running time of $O(1)$ per operation to insert x into S, delete x from S, and check whether $x \in S$ or not, respectively. Explain how such a data structure exists or why it does not exist.
(f) (**T** , **F**) Checking whether there is a pair of non-equal elements in an array of n real numbers requires $\Omega(n \lg n)$ time. (Note: you are NOT allowed to use any linear-time sorting algorithm.)

6. (6%) Compute the asymptotic tight bound (i.e .the Θ-notation) for $T(n) = 2T(\frac{n}{2}) + n \lg n$ in terms of n and $\lg n$.

[!!Note!! Remember to Solve Problem 7. in Next Page]

國立成功大學九十六學年度碩士班招生考試試題

編號： 381 系所：資訊管理研究所乙組 科目：資料結構

本試題是否可以使用計算機： □可使用 ， ☑不可使用 （請命題老師勾選）

7. (20%) Given a connected undirected graph $G = (N, A)$ where N and A denote the set of the nodes and arcs in G, respectively. Let $|N| = n$ and $|A| = m$. Let c_{ij} denote the length for arc (i, j) and $c_{ij} > 0$ for each $(i, j) \in A$. Answer the following questions. (Answers without explanation get at most 1 point.)

(a) [4%] Suppose $c_{ij} = w > 0$ for each $(i, j) \in A$. Can you give a method faster than the Dijkstra's algorithm to calculate the shortest path length from a node $s \in N$ to a node $t \in N$? If **yes**, explain how and the complexity of your method in terms of n, m, or w; **otherwise**, explain why no such a method exists.

(b) [16%] Suppose we store the arc lengths by an $n \times n$ adjacency matrix C, where $C[i][j] = c_{ij}$ for each $(i, j) \in A$, $C[i][j] = M$ for each $(i, j) \notin A$, and M represents a very large number. Each node $i \in N$ is associated with a distance label $d[i]$, which is set to be M in the beginning. Let P and Q denote two node sets such that $P \cup Q = N$. Set $P = \emptyset$ and $Q = N$ in the beginning. Given an origin node s and destination node t, then $d[s] = 0$. The Dijkstra's algorithm iteratively conducts two operations: **NodeSelection**(Q) and **DistanceUpdate**(d, C). In particular, **NodeSelection**(Q) selects a node i in Q with the minimum distance label and moves i from Q to P; and **DistanceUpdate**(i, d, C) updates $d[j] = \min_{(i,j)\in A}\{d[j], d[i] + C[i][j]\}$.

Algorithm Dijkstra(P, Q, d, C)
initialize: read s, t; set $P = \emptyset$, $Q = N$ and $d[\cdot] = M$ except $d[s] = 0$;
while $t \in Q$ **do**
    $i =$ **NodeSelection**(Q);
    DistanceUpdate(i, d, C);
end while

(b1) [4%] Suppose we use a linked-list to store P and Q. What is the OVERALL complexity in terms of n and m for **NodeSelection**(Q) and **DistanceUpdate**(i, d, C), respectively? (In other words, how much time does these two operations spend in the entire algorithm for a given s and t?)
(b2) [4%] Answer the same questions as (b1), if we use a linked-list to store P but a min-heap to store Q.
(b3) [4%] Instead of using an $n \times n$ adjacency matrix C to store the arc length, suppose we use $W[k]$ to store the arc length for each arc $k \in A$. Also, suppose we use a linked list $L[i]$ for each node $i \in N$ to store the indices of the arcs outgoing from node i. Thus **DistanceUpdate**(i, d, W, L) updates $d[j] = \min_{(i,j)\in L[i] \text{ with an arc index } k}\{d[j], d[i] + W[k]\}$. What is the OVERALL complexity in terms of n and m for **NodeSelection**(Q) and **DistanceUpdate**(i, d, W, L), respectively if we use a linked-list to store both P and Q?
(b4) [4%] Answer the same questions as (b3), if we use a linked-list to store P but a min-heap to store Q?