※ 考生請注意：**本試題不可使用計算機**

※ 本試題共計 3 頁。作答時可不必抄題，但請務必於答案卷將各作答題之題號標註清楚。

※ Note that, throughout this paper, the height of a tree with only one node is defined as 1 while an empty tree has height of zero.

1. 名詞或術語解釋 (10 分；請解釋下列各題之名詞或術語；每題 2 分)

(1-1) Huffman Tree (1-2) Rehashing (1-3) Adjacency multilist

(1-4) ω-notation (1-5) Sollin's algorithm

2. 是非題 [A] (10 分；下列 (2-1) ~ (2-5) 各題敘述若正確請以 **T** 表示，若錯誤請以 **F** 表示；每題 2 分)
For each statement in (2-1) ~ (2-5), please indicate **T** if it is correct and indicate **F** otherwise. (2 points each)

(2-1) If the balance factor of a node in an AVL tree is 1, then the balance factor of its left child must be 0.

(2-2) An algorithm of time complexity $O(n^2)$ is always more efficient than the one of time complexity $\Theta(n^2)$.

(2-3) When n ($n > 2$) data items are sorted using bubble sort algorithm and the number of key comparison is k, if the number of data item exchanges is 1, then $k \le 2 \cdot (n-1)$.

(2-4) An algorithm of the divide-and-conquer paradigm can usually be implemented using recursion.

(2-5) If T is a 2-3 tree of n nodes and h is the height of T, then $n \ge 2^h - 1$.

3. 是非題 [B] (20 分；下列 (3-1) ~ (3-10) 各題敘述若正確請以 **T** 表示，若錯誤請以 **F** 表示；每題 2 分，答對得 2 分，答錯得 0 分並倒扣 1 分，未作答得 0 分)
For each statement in (3-1) ~ (3-10), please indicate T or O if it is correct and indicate F or メ otherwise.
(You get 2 points for each correct answer and lose 1 point for each incorrect answer.)

(3-1) Using a stack can help convert an infix expression into prefix expression.

(3-2) The quick sort algorithm can be implemented to do external sorting with the same time complexity.

(3-3) In static hashing, if the value of the key density is equal to the value of the hash table's loading factor, then there will be no overflow.

(3-4) For any non-empty binary search tree T, if n_0 is the number of leaf nodes and n_2 is the number of nodes with two children nodes, then $n_0 \ge n_2$.

(3-5) For an undirected non-multi-graph G = (V, E), where V is the set of vertices and E is the set of edges. If h_G stands for the height of G's spanning tree, then $\log_2|V| \le h_G \le |V|$.

(3-6) If the element A[3][11] of an integer array A[m][n] is stored at the memory location of address 6774 and the address of element A[8][2] is stored at 7078, then each element of A is allocated 4 bytes.

(3-7) For every vertex v_i in a connected digraph G, the in-degree of v_i is equal to its out-degree.

(3-8) A network can be represented using an adjacency matrix.

(3-9) The height of a red-black tree with n internal nodes is at least $2 \cdot \log_2(n+1)$.

(3-10) A max binomial heap is also a connected graph.

(背面仍有題目,請繼續作答)

※ 考生請注意：**本試題不可使用計算機**

4. 選擇題 (30 分，每小題 3 分)

For each question from (4-1) to (4-10), please choose all appropriate items from the ones given in the box below as your answer. Note that (i) there might be more than one appropriate item and you must give all such items to get the points; (ii) all the items in the box can be chosen for more than one question; (iii) the item (A) and (a) are different items. In case that you think none of the items in the box can be used as an answer, please mark ○ as your answer. (3 points each)

(A) Radix sort	**(B)** Quick sort	**(C)** Selection sort
(D) Topological sort	**(E)** Binary tree sort	**(F)** Merge sort
(G) Bubble sort	**(H)** Interval heap	**(J)** Insertion sort
(K) Queue	**(L)** Heap	**(M)** Red-Black tree
(N) Stack	**(P)** Splay tree	**(Q)** Binary tries
(R) LL rotation	**(S)** LRr operation	**(T)** Leftist tree
(U) RR rotation	**(V)** LLr operation	**(W)** Spanning tree
(X) RL rotation	**(Y)** B-Tree	**(Z)** AVL tree
(a) $O(1)$	**(b)** $O(n)$	**(c)** $O(n^2)$
(d) $O(n \cdot \log n)$	**(e)** $O(\log n)$	**(f)** $\Omega(2^n)$
(g) $O(n^m)$	**(h)** Division	**(j)** Mid-square
(k) $O(n!)$	**(m)** Folding	**(n)** LR rotation
① 1	② 2	③ 3
④ 4	⑤ 5	⑥ 6
⑦ 7	⑧ 8	⑨ 9

(4-1) Which might be the time complexity in solving an NP-Complete problem?

(4-2) Given a sequence of n numbers and $n > 10000$, the total number of comparisons in a operation of searching 10 numbers among the n numbers is at least __(4-2)__ in terms of big O notation.

(4-3) If the number sequence "6, 3, 1, 4, 7, 8, 7, 2, 8, 9, 5" is input to build an AVL tree, then the right child of the root node is __(4-3)__.

(4-4) Which data structure is explicitly used in general when performing a level-order traversal of a binary tree and the BFS operation of visiting all nodes of a graph.

(4-5) Suppose during an operation, a sequence of n elements are sorted into non-ascending order using quick sort algorithm and then sorted again into non-descending order using the x algorithm. If the total time complexity of the whole operation is $O(n \cdot \log n)$, then which algorithm(s) might x be?

(4-6) What is the worst-case time complexity of searching with an optimal binary search tree of n nodes?

(4-7) If the result of doing in-order traversal on a binary tree T is "1, 2, 3, 4, 5, 6, 7, 8, 9" and the left subtree of T's root node x is a complete binary tree with root node 4, then the right child node of x is __(4-7)__.

(4-8) Which operation(s) can be used in designing a hash function?

(4-9) Which operation(s) will be used after deleting a node from an AVL tree?

(4-10) Which data structure(s) is(are) height-balanced search tree(s)?

※ 考生請注意：**本試題不可使用計算機**

5. Given an arbitrary binary tree T of n nodes, please do the following:

(5-1) Develop an algorithm to find the height of T. Please describe your algorithm using *pseudo code*. (8 points).

(5-2) Analyze and give the time complexity of your algorithm described in (5-1) in terms of asymptotic notations. (5 points)

(5-3) Analyze and give the space complexity of your algorithm described in (5-1) in terms of asymptotic notations. (5 points)

(5-4) In implementing your algorithm given in (5-1) using a high-level programming language, please describe your program using flow chart notations. (5 points).

(5-5) If the data stored in the nodes of T are date (including year, month, and day) and time (including hour, minute, and second), please present the segment of program code using C or C++, with optimization in memory consumption, to implement the node of T. (5 points)

(5-6) On a computer with 32-bit CPU and 2GB memory, how many bytes are consumed by each node of T in your implementation given in (5-5). (2 points)