※ 注意：請用 2B 鉛筆作答於答案卡，並先詳閱答案卡上之「畫記說明」。

是非題 (共 25 小題，若你覺得該小題命題正確，請填 (A)，錯誤，請填 (B)。每題答對得 4 分，答錯倒扣 4 分，未填答者不計分也不扣分，倒扣至總分為 0 分為止)

1. The time complexity to find an arbitrary element in a singly linked list is O(n), while the time complexity to find an arbitrary element in an array is O(log n), where n is the number of elements in the list and array.

2. The memory usage for a doubly linked list is $\Theta(n^2)$, where n is the number of elements in the list.

3. The time complexity for the following code is $\Theta(n \cdot n!)$.

```
void permuteGen(char* a, const int k, const int n)
{
    if (k == n - 1) {
        for (int i = 0; i < n; i++)
            cout << a[i] << " ";
        cout << endl;
    }
    else {
        for (int i = k; i < n; i++) {
            swap(a[k], a[i]);
            permuteGen(a, k + 1, n);
            swap(a[k], a[i]);
        }
    }
}
```

4. Given a balanced binary tree where, for any arbitrary internal node, the numbers of nodes in its left and right sub-trees diff for at most one node. The time complexity to find an element in this tree is O(log n), where n is the number of elements.

5. Let f(n) be a linear function of n. Then $f(n) = \Omega(\log^k n)$ for any power of k.

6. Given a static (i.e. fixed size) array, the time complexity to remove an arbitrary element from the array can be O(1).

7. A k-nary tree is a tree in which each node has at most *k* children. Given a k-nary tree with *n* nodes, the number of edges is *n*-*k*+1.

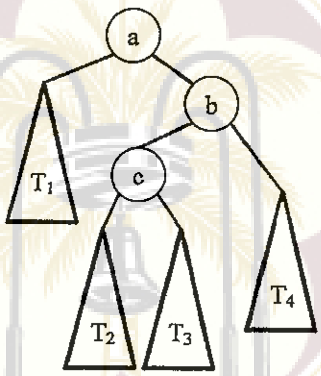
見背面

8. Given a k-nary tree with $n$ nodes, the height of the tree is at least $\log_k n - 1$.

9. Given a complete binary tree. Let's perform the post-order traversal from the root and number the nodes with numbers starting from 1. For a node with number k, its parent must be numbered as 2k or 2k+1.

10. The height of an AVL tree is bounded by O(log n), where n is the number of nodes in the tree.

11. Given an AVL tree as shown below, let $T_1$, $T_2$, $T_3$ and $T_4$ be sub-trees with the same height. If we insert a new node to the leaf of $T_3$, then node c will be the root of the AVL tree after rotations.




12. It takes O(log n) time to find the minimal node (i.e. the node with the smallest value) of an AVL tree.

13. Inserting nodes to a blank red-black tree with the following order: { 100, 70, 50, 40, 30, 20, 10 }. The root node of the resulted tree will be 40.

14. For the resulted red-black tree of the previous problem, the black height (i.e. the number of black nodes in each path from root to leaf) is 2.

15. For any node in a red-black tree, the difference of heights between its left and right sub-trees must be smaller than or equal to 2.

16. A B-tree of order 2 is a full binary tree.

17. Inserting nodes to a 2-3-4 tree with the following order using top-down insertion: { 10, 9, 8, 7, 6, 5, 4, 3, 2, 1 }. The root node of the resulted tree will be a 2-node.

18. The height of a 2-3-4 tree must be smaller than or equal to the height of a 2-3 tree with the same data.

接次頁

19. When inserting a set of data into a binominal heap, the sizes of its binominal trees are independent of the data insertion order.

20. The smallest element of a binominal heap must be the root of the binominal tree with the maximum number of nodes.

21. Given a directed acyclic graph (DAG) with lists of source and sink nodes. For two nodes *u* and *v* that are adjacent, we call *u* as *v*'s parent node if there is an edge from *u* to *v*. If we number the nodes, from smallest to largest, by performing the post-order traversal from the source node, then every node must have the number that is greater than its parent.

22. The complement graph of a non-empty complete graph must contain at least one clique.

23. Given an arbitrary directed graph, it takes $O(2^n)$ time to determine whether there exists a path between two nodes, where n is the number of nodes.

24. A bipartite graph must be two-colorable.

25. Let f(n) be the Fibonacci series where f(0) = 0 and f(1) = 1. Let H be a hash with 10 buckets, and let |H(i)| denote the number of elements in the i's bucket, for i = 0 to 9. If we insert the first 12 numbers of Fibonacci series, that is, f(0) to f(11), into the hash H with the following hash function "$h(n) = n^2 \% 10$". Then |H(4)| > |H(5)|.

試題隨卷繳回