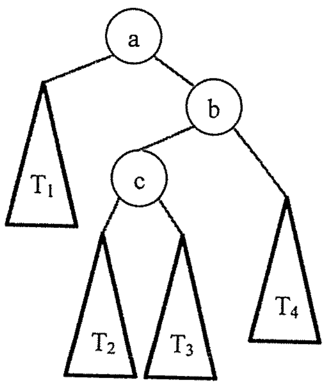
☑ 答案請務必填在電腦閱卷之「答案卡」上。

(一) 是非題 (共 10 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，倒扣至是非題總分為 0 分為止)

1. Let f(n) be the time complexity of the graph coloring problem (i.e. two adjacent nodes cannot have the same color), where n is the number of nodes of the graph. Then $f(n) = \Omega(n^k)$ for any power of k.

2. Given an AVL tree as shown below, let T_1, T_2, T_3 and T_4 be sub-trees with heights equal to h, h-1, h-1, and h, respectively. If we insert a new node to the leaf of T_2, then node a will be the left child of node b after rotation(s).



3. It takes O(log n) time to find the minimal data in a hash, where n is the number of data in the hash.

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

5. If the "insert" operation of a Fibonacci heap is rarely called, the amortized time complexity of the heap operations can be O(1).

6. The "dominator" d of a node v in a directed acyclic graph (DAG) is a node that all the paths going from v to any of the sink nodes must go through d. Let n be the number of nodes in the graph, then finding the set of dominators for all nodes in an arbitrary DAG can be O(n).

7. For any node in a 2-3-4 tree, the difference of heights between its left and right sub-trees can be smaller than or equal to 1.

8. Even though a leftist heap is very unbalanced, the time complexity for its "insert" and "delete" operations can still be O(log n), where n is the number of nodes in the heap.

9. The cardinality of a cut-set in the complete graph of n nodes must be no less than n-1.

10. Inserting nodes to a 2-3 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.

見背面

(二) 複選題 (共六小題，每題 10 分, 題內每個選項單獨計分，答對得 2 分，答錯倒扣 2 分，倒扣至複選題總分為 0 分為止。如某題未作答則該題得 0 分，不到扣。)
(例：正確答案為 ABC，若答 BCD，則答對三個選項(BCE)，答錯兩個選項(AD)，故共計得 2 分)

11. For the singly linked list with the following structure,

```
class List {
    ListNode   *_first; // first element in the list
};
class ListNode {
    int          _data;
    ListNode    *_next;  // pointer to the next ListNode
};
```

Assume that whenever a number is to be inserted into or deleted from the list, it must be operated properly so that all the numbers in the list are always sorted ascendingly. Which of the following statement(s) is(are) NOT correct?

(A) The time complexity of inserting a number that is larger than all the numbers in the list is higher than that of inserting a number that is smaller than all the numbers, because there is no data member "`_last`" that records the largest number.

(B) If we add a data member "`_last`" that records the largest number in the list, the time complexity of deleting the largest number will be the same as deleting the smallest number.

(C) Given two linked lists of the above type, it takes O(n^2) to merge these two lists into a new sorted list, where n is the number of nodes of the new list.

(D) Given a linked list of the above type, its deletion operation can be O(log n), were n is the number of nodes of the list.

(E) The complexity of computing the size of the list is the same as that of checking whether a list is empty.

12. Which of the following is(are) true for a Binomial Heap/Tree?

(A) The time complexity to find the minimum node in a Binomial Heap of n nodes is O(1).

(B) The height of a Binomial Heap of n nodes can be O(n).

(C) A Binomial Heap of n nodes contains O(log n) Binomial Trees.

(D) It takes O(log n) time to compose two Binomial Trees into a Binomial Tree of n nodes.

(E) Given a Binomial Heap of k Binomial Trees, after deleting the minimum node, this Heap may contain more than k Binomial Trees.

接次頁

13. Given a hash of strings with number of buckets equal to 100 and hash function defined as:

```
int hashFunction(const string& str)
{
   int t = 0;
   for (int i = 0; i < str.length(); i++)
      t += (str[i] << (i*2));
   return (t % 100);
}
```

where the function "length()" returns the length (i.e. the number of characters) of the string, and the operator [i] returns the i^th character in the string (note: index starting from 0). FYI, the ASCII codes for the numbers 0~9, letters A~Z and a~z, are 48~57, 65~73, 97~122, respectively. Let the function "`int bucketNumber(const string& str)`" return the index of bucket for the string "`str`" in the hash (i.e. between 0 and 99). For example, `bucketNumber("a")` should return 97.

Which of the following statement(s) is(are) true?

(A) `bucketNumber("bj4")` returns 56.
(B) `bucketNumber("abc")` is smaller than `bucketNumber("xyz")`.
(C) `bucketNumber("abc")` is smaller than `bucketNumber("ABC")`.
(D) Inserting `"a"` and `"add"` will introduce a collision.
(E) If we change the number of buckets to 200, the truthfulness of the above 4 statements remains the same since the hash function does not change.

14. Let T_a and T_b be two balanced binary search trees with roots r_a and r_b, numbers of nodes n_a and n_b, and heights h_a and h_b, respectively. Assume the values of nodes in T_a are smaller than any of the nodes in T_b. Let's compose a new tree T_c by creating a new node r_c with two children r_a and r_b (i.e. any node in $T_a < r_c <$ any node in T_b). Which of the following is(are) true about the "tree" data structure?

(A) T_c is also a balanced binary search tree.
(B) The number of edges in T_c is $(n_a + n_b - 1)$.
(C) The height of T_c is $\max(h_a, h_b) + 1$.
(D) The number of paths from r_c to the leaf nodes is $O(\log(\max(n_a, n_b)))$.
(E) It takes $O(\log n_a)$ time to find the minimum node.

15. Which of the following statement(s) about "tree" is(are) true?

(A) Let a "complete tree" be a special case of a tree, in which all the levels, except perhaps the last, are *full*; while on the last level, any missing nodes are to the right of all the nodes that are present. Given a complete tree in which the number of children of a *full* node at level k be $(k+2)$. Then the number of nodes for such a complete tree of height h is $O(2^n)$ (i.e. exponential).

(B) For a binary search tree, the post-order search procedure can visit the nodes from the smallest to the largest.

(C) For an AVL tree with number of nodes equal to 7, its possible maximum height is 3.

(D) The probability of getting a random binary tree with minimum height (i.e. height = $\log_2 n$, where n is the number of nodes in the tree) is greater than 50%, for n is large enough.

(E) For a red-black tree that is full, let its number of red nodes be r. Then its number of black nodes must be $(2*r + 1)$.

16. Which of the following statement(s) about "graph" is(are) true?

(A) A plannar graph must be a bipartite graph.

(B) The cardinality of the largest clique in an octahedron is 3.

(C) For a non-empty directed graph, there must exsit one and only one strongly connected component (SCC).

(D) The number of paths in a directed acyclic graph (DAG) can be exponential to the number of nodes in the graph.

(E) Let the maximum number of edges of a node in a graph be e. Then the cardinality of the minimum clique partition (i.e. the clique covering number) must be greater than or equal to $n/(e+1)$, where n is the number of nodes in the graph.

試題隨卷繳回