台灣聯合大學系統114學年度碩士班招生考試試題

類組：電機類　科目：資料結構(3002)　　共 12 頁 第 1 頁

一、單選題 1-20，每題 2 分，答錯不倒扣

1. for(int i=0; i*i < n; i++) {statement… }. Time complexity of the for-loop is
   (A) $O(1)$
   (B) $O(n)$
   (C) $O(n^2)$
   (D) $O(n^{1/2})$

2. Which of the following data structure that stores data in a first-in, last-out order?
   (A) Stack
   (B) Queue
   (C) Tree
   (D) List

3. The postfix expression of A*(B+C)-D is:
   (A) A*(BC+)-D
   (B) ABC+*D-
   (C) -*A(+BC)D
   (D) None of the above

4. Which of the following sorting algorithms has an average time complexity of O(nlogn)?
   (A) Bubble sort
   (B) Selection sort
   (C) Heap sort
   (D) Insertion sort

5. Which of the following sorting algorithms is NOT in-place?
   (A) Quick sort
   (B) Merge sort
   (C) Heap sort
   (D) All of the above are in-place

注意：背面有試題

6. If you want to use a heap structure to keep the top-k numbers by scanning through a list of numbers, what is the right choice?
   (A) min-heap
   (B) max-heap
   (C) Both will work equally well
   (D) Trees perform better

7. What is the algorithm's computation efficiency using a heap sized k (given in the previous question) to maintain top-k numbers by scanning through a list of numbers?
   (A) O(n)
   (B) O(nlogn)
   (C) O(nlogk)
   (D) O(k)

8. The height of a tree is defined as the number of edges present in the longest path connecting the root to a leaf node. What is the height of a max binary heap after inserting 3, 41, 52, 26, 38, 57, 9, 49 sequentially? The heap is empty initially.
   (A) 1
   (B) 2
   (C) 3
   (D) 4

9. What is the height of a binary search tree after inserting 3, 41, 52, 26, 38, 57, 9, 49 sequentially? The tree is empty initially.
   (A) 1
   (B) 2
   (C) 3
   (D) 4

10. Which of the following arrays is NOT a max binary heap?

(A) {16, 14, 9, 4, 8, 12, 5, 3, 2}

(B) {16, 14, 9, 4, 12, 8, 5, 3, 2}

(C) {16, 8, 14, 5, 2, 9, 12, 4, 3}

(D) {16, 9, 14, 8, 5, 12, 2, 4, 3}

11. Suppose that we have numbers between 1 and 1000 in a binary search tree and we want to search the number 101. Which of the following sequences could possibly be the sequence of nodes examined?

(A) 20, 252, 41,49, 130, 120, 131, 101
(B) 252, 42, 79, 200, 190, 86, 130, 121, 101
(C) 80, 252, 90, 130, 120, 125, 110, 101
(D) 90, 252, 100, 131, 130, 92, 110, 101

12. Five people A, B, C, D, and E are standing in a queue. A is standing next to C. D and E are not standing next to each other. Who is NOT possibly standing third in the queue?

(A) B

(B) D

(C) E

(D) All of the above

13. Let Fibonacci(0) = 0, Fibonacci(1) =1, Fibonacci(2) = 1, Fibonacci(3) = 2, Fibonacci(4) = 3, and Fibonacci(i+2) = Fibonacci(i) + Fibonacci(i+1), what is Fibonacci(7) equal to?

(A) 34

(B) 21

(C) 8

(D) 13

注意：背面有試題

Answer questions 14-16 according to the following C code.

```
struct Node {
    int data;
    struct Node* next;
};
int funcA(struct Node* head) {
    struct Node *slow = head, *fast = head;
    while (slow && fast && fast->next) {
        slow = slow->next;
        fast = fast->next->next;
        if (slow == fast) {
            return 1;
        }
    }
    return 0;
}
```

14. What is the purpose of funcA()?

(A) Find repeated numbers in a linked list

(B) Find the unique numbers in a linked list

(C) Find a loop in a linked list

(D) Reverse a linked list

15. Time complexity of funcA()?

(A) O(1)

(B) O(n)

(C) $O(n^2)$

(D) O(nlogn)

16. Auxiliary memory space required by funcA()?

(A) O(1)

(B) O(n)

(C) $O(n^2)$

(D) O(nlogn)

注意：背面有試題

Answer questions 17-20 according to the following C++-style pseudo-code.

```
int funcB(string s) {
    stack<int> st;
    st.push(-1);
    int maxLen = 0;
    for (int i = 0; i < s.length(); i++) {
        if (s[i] == '(') {
            st.push(i);
        } else {
            st.pop();
            if (st.empty()) {
                st.push(i);
            } else {
                maxLen = max(maxLen, i – st.top());
            }
        }
    }
    return maxLen;
}
```

17. Let s = “()”, funcB(s) will return
    (A) 0
    (B) 2
    (C) 4
    (D) 6

18. Let s = “()(())”, funcB(s) will return
    (A) 0
    (B) 2
    (C) 4
    (D) 6

19. Which of the following statements is NOT correct? Assuming s only consists of '(' and ')'.
   (A) For every opening parenthesis, '(', we push its index onto the stack
   (B) For every closing parenthesis, ')', we pop the stack
   (C) If the stack becomes empty after popping, it means we've encountered an unmatched closing parenthesis ')'
   (D) None of the above is correct

20. What is the purpose of funcB()?
   (A) Find the length of the input string
   (B) Find the maximum length of valid parentheses, having matched '(' and ')'
   (C) Check if given parentheses expression is balanced or not
   (D) Reverse the input string

二、複選題 21-24，每題 5 分，答錯要倒扣。（答錯倒扣 1 分，倒扣至本大題(複選題)0 分為止。）

21. A version of quicksort is given in the following C code.

```
void quicksort(int v[], int left, int right) {
    int pivot, i, j, tmp;
    if (left < right) {
        i = left;
        j = right + 1;
        pivot = v[left]; // used as a pivot value
        do {
            do i++; while (v[i] < pivot);
            do j--; while (v[j] > pivot);
            if (i < j) {
                tmp = v[i];
                v[i] = v[j];
                v[j] = tmp;
            }
        } while (i<j);
        tmp = v[left];
        v[left] = v[j];
        v[j] = tmp;
        quicksort(v, left, j-1);
        quicksort(v, j+1, right)
    }
}
int main() {
    int V[] = {12,2,16,30,8,28,4,10,20,6};
    quicksort(V,0,9);
}
```

Which of the following statements are true: (5%) multiple anwsers

(A) The quicksort() function sorts the given array in a non-increasing order.

(B) The quicksort() function is a stable sorting algorithm.

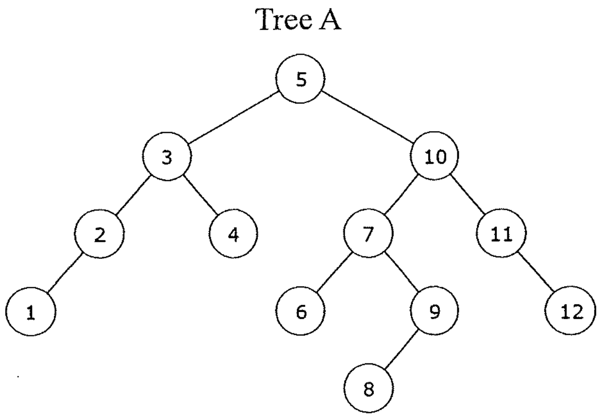
(C) The quicksort() function is an in-place sorting algorithm

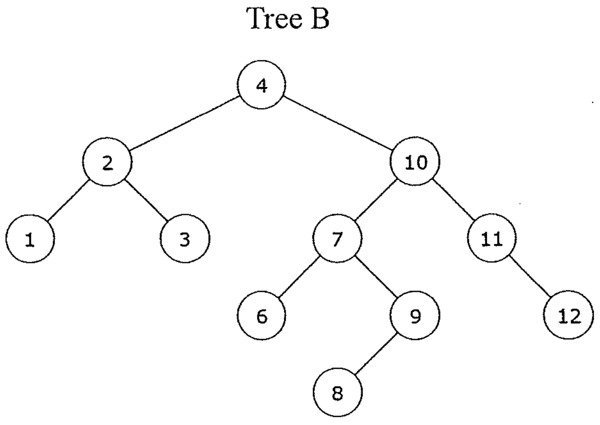
(D) 28 is used as a pivot value during the sorting process

(E) 8 is used as a pivot value during the sorting process

注意：背面有試題

22. Height: Assume that the height of a tree node is the number of edges on the longest path from the node to a leaf. A leaf node will have a height of 0. Balance: The balance of a node is defined: left_child_node's height - right_child_node's height. The balance of a leaf node is 0 (since The height of a "None" child node is defined as -1).





Which of the following statements are true based on the definitions given above. (5%) multiple anwsers

(A) The balance of every node in Tree A is 1, 0, or -1.

(B) The balance of every node in Tree B is 1, 0, or -1.

(C) Tree A is a binary search tree

(D) Tree B is a binary search tree

(E) Both Tree A and Tree B are AVL trees.

23. Given the following directed graph:

Let G = (V, E) be a graph, V and E are sets of vertices and edges.

V = {V0, V1, V2, V3, V4, V5, V6}

E = {(V0->V1), (V0->V3), (V1->V3), (V1->V4), (V2->V0), (V2->V3), (V2->V5), (V3->V2), (V3->V4), (V3->V5), (V3->V6), (V4->V6), (V5->V6), (V6->V4), (V6->V5)}, where (V0->V1) represents an edge from V0 to V1.

Which of the following statements are true based on the definitions given above. (5%) multiple anwsers

(A) G is a directed acyclic graph

(B) Do a breadth-first search beginning at V0, V0>V1>V2>V3>V4>V5>V6 is a possible order visited by the algorithm

(C) Do a depth-first search beginning at V3, V3>V6>V4>V5>V2>V0>V1 is a possible order visited by the algorithm

(D) Do a breadth-first search beginning at V2, V2>V0>V3>V5>V1>V4>V6 is a possible order visited by the algorithm

(E) Do a depth-first search beginning at V1, V1>V3>V2>V5>V>V4>V0 is a possible order visited by the algorithm

注意：背面有試題

24. We have an open addressing hash table. Please simulate the hash table's behavior storing integers given the following conditions:

✓The hash table size is 9.

✓It uses quadratic probing for collision resolution.

✓The hash function uses the int value (plus any probing needed) mod the size of the table as described in the following.

Hashing+Probing: ((k mod m) + i^2) mod m, k is key, m is the size of the table, probing from i = 0, 1, 2,…

What values will be in the hash table after the following sequence of insertions?

18, 16, 10, 7, 26

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]

Which of the following statements are correct. (5%) multiple anwsers

(A) 18 is stored at [0]

(B) 10 is stored at [1]

(C) 26 is stored at [8]

(D) 7 is stored at [7]

(E) 16 is stored at [7]

三、問答題 1-4 題

1. Consider the following function, where n is a positive integer:

```
Function mystery(n):
    If n <= 1:
        Return 1
    Else:
        Return mystery(n / 2) + mystery(n / 2) + n
```

Please (A) answer the time complexity of this function (4 points) and (B) justify your answer with a detailed explanation (6 points).

2. The following table is the adjacency list and edge weights of a graph:

Node	Adjacent Nodes (Target Node, Weight)
0	(1, 4), (2, 1)
1	(3, 1)
2	(1, 2), (3, 5)
3	

(A) Please draw the graph based on the table. (5 point)

(B) Using Dijkstra's algorithm, calculate the shortest path from node 0 to all other nodes and draw the resulting shortest path. (5 point)

注意：背面有試題

3. Based on the following code, please answer the questions below:

(A) Why can the hasCycle function successfully detect the presence of a cycle in the linked list when link node4.next to node2? Please explain in detail how the slow and fast pointers work and at which node they meet. (5 point)

(B) If the assignment link node4.next to node2 is removed and the program is executed, what will be the output of the hasCycle function? Please explain the reason for this situation and how the algorithm handles it. (5 point)

```
Class ListNode
    Attributes: value, next

    Constructor(value):
        value = value
        next = null

Function hasCycle(head):
    If head is null or head.next is null:
        Return false

    slowPointer = head
    fastPointer = head

    While fastPointer is not null and fastPointer.next is not null:
        slowPointer = slowPointer.next
        fastPointer = fastPointer.next.next

        If slowPointer == fastPointer:
            Return true

    Return false

Main:
    Create node1 with value 1
    Create node2 with value 2
    Create node3 with value 3
    Create node4 with value 4

    Link node1.next to node2
    Link node2.next to node3
    Link node3.next to node4
    Link node4.next to node2

    result = hasCycle(node1)
    Print "Does the linked list have a cycle? " + result
```

注意：背面有試題

4. Based on the following codes, please answer the questions below:
   (A) What is the tree structure in the code? (3 points)
   (B) What is the output of the code? (3points)
   (C) In the zigzagLevelOrder method, if the initial value of leftToRight is changed to false, what will the output be after execution? Please provide a detailed explanation. (4 points)

```
Class TreeNode
    Attributes: value, left, right

    Constructor(value):
        value = value
        left = null
        right = null

Function zigzagLevelOrder(root):
    result = empty list of lists
    If root is null:
        Return result

    queue = empty queue
    Enqueue(root, queue)
    leftToRight = true

    While queue is not empty:
        levelSize = size of queue
        currentLevel = empty list

        For i = 0 to levelSize - 1:
            currentNode = Dequeue(queue)

            If leftToRight:
                Append(currentNode.value, currentLevel)
            Else:
                Insert at beginning(currentNode.value, currentLevel)

            If currentNode.left is not null:
                Enqueue(currentNode.left, queue)
            If currentNode.right is not null:
                Enqueue(currentNode.right, queue)

        Append(currentLevel, result)
        leftToRight = not leftToRight

    Return result

Function main:
    root = new TreeNode(1)
    root.left = new TreeNode(2)
    root.right = new TreeNode(3)
    root.left.left = new TreeNode(4)
    root.left.right = new TreeNode(5)
    root.right.right = new TreeNode(6)

    result = zigzagLevelOrder(root)
    Print(result)
```