考 試 科 目	程式設計與資料結構	所 別	數位內容碩士學位學程/資訊技術組 5152	考 試 時 間	3 月 7 日(日) 第 4 節

可用中文或英文回答

1. (20%) True or False (Use O for true or X for false as the answer to each of the following statements.)

(1) $O(n^d)$ is of exponential order, where d is a constant.

(2) All recursive procedures can be implemented as iterative procedures.

(3) Some sorting algorithms can perform in linear time.

(4) A binary search in a linear sorted array needs $O(\log n)$ time.

(5) With the preorder traversal in a binary search tree, we can get a sorted sequence.

(6) When some rotations are done on an AVL tree due to an unbalanced node after insertion, the height of the tree may be changed.

(7) The number of elements stored in the root node of a B-tree of order n is between $\left\lfloor \frac{n-1}{2} \right\rfloor$ and n-1.

(8) All leaves in a B-tree are on the same level.

(9) The load factor for a hash table is defined as the ratio of the number of elements in the hash table to the table size.

(10) For all possible inputs, a linear algorithm to solve a problem always perform faster than a quadratic algorithm to solve the same problem.

2. (20%) Single Selection

(1) Suppose that n is the number of input data. Which statement is correct?

(A) A binary search in a linear sorted array needs $O(\log n)$ time.

(B) An insertion in a linear sorted array needs $O(\log n)$ time.

(C) The binary search can be performed in a linearly linked list if the list is sorted.

(D) A deletion in a linearly (singly) linked list can be done in $O(1)$ time if the node to be deleted is known.

(2) Which sorting algorithm(s) has time complexity $O(n \log n)$ in the worst case?

(A) Bubble sort

(B) Shell sort

(C) Heapsort

(D) Quicksort.

(3) Which statement is *correct* for an AVL tree?

(A) An insertion requires at most three rotations.

(B) A deletion requires at most two rotations.

(C) The level difference of any pair of leaves is at most one.

(D) The height difference of the two subtrees of any node is at most one.

備註	試 題 隨 卷 繳 交

考試科目	程式設計與資料結構	所別	數位內容碩士學位學程/資訊技術組 5152	考試時間	3月7日(日) 第4節

(4) Which statement is *correct* for an almost complete binary tree?
(A) It is a strictly binary tree.
(B) Each node, except for the root, has its sibling node.
(C) At each node, the height of the left subtree is always greater than or equal to that of the right subtree.
(D) All leaf nodes can appear only on the lowest level.

(5) If G is a directed graph with 20 vertices, how many boolean values will be needed to represent G using an adjacency matrix?
(A) 40
(B) 200
(C) 210
(D) 400

(6) How many linked lists are needed to represent a graph with n nodes and m edges, when using an edge list representation?
(A) m
(B) n
(C) $m + n$
(D) $m*n$

(7) An array of queues can be used to implement a priority queue, with each possible priority corresponding to its own element in the array. When is this implementation NOT feasible?
(A) When the number of possible priorities is huge.
(B) When the number of possible priorities is small.
(C) When the queues are implemented using a linked list.
(D) When the queues are implemented with circular arrays.

(8) Suppose we are sorting an array of eight integers using quicksort, and we have just finished the first partitioning with the array looking like this:

2 5 1 7 9 12 11 10

Which statement is correct?
(A) The pivot could be either the 7 or the 9.
(B) The pivot could be the 7, but it is not the 9.
(C) The pivot is not the 7, but it could be the 9.
(D) Neither the 7 nor the 9 is the pivot.

(9) Which of these is the correct big-O expression for 1+2+3+...+n?
(A) O(log n)
(B) O(n)
(C) O(n log n)

備註	試題隨卷繳交

考試科目	程式設計與資料結構	所別	數位內容碩士學位學程/資訊技術組 5152	考試時間	3月7日(日) 第4節

(D) $O(n^2)$

(10) Consider a complete binary tree with 1000 nodes, with the data stored in the elements *a*[0]..*a*[999] of an array. Where are the parent and the right child of the node stored at *a*[100]? (Assume that the left child is stored before the right child.)

(A) Parent at 49, right child at 200

(B) Parent at 49, right child at 201

(C) Parent at 50, right child at 200

(D) Parent at 50, right child at 201

3. **(10 %)** A function *f* in the C language is defined as follows.

```
int f(int a, int m) {
    int t, ans;
    ans=1;
    t=a;
    while(m>0) {
        if (m % 2 ==1) /* e.g. 7%2=1, 6%2=0 */
            ans = ans * t;
        t = t * t;
        m = (int) (m / 2); /* e.g. 7/2=3, 6/2=3 */
    } /* end of while() */
    return(ans);
}
```

(a) What is the answer of `f(3,5)`?

(b) Use arithmetic expressions or simple sentences to describe what *f* is.

4. **(8 %)** An array *int A*[*m*][*n*] is given. Assume that each element of array *A* occupies 4 units of storage. Suppose the addresses of *A*[3][4] and *A*[2][7] are 404 and 544, respectively. The first element of *A* is *A*[0][0].

(a) Is array *A* in *row-major* or *column-major*? Why?

(b) What is the address of *A*[0][0]?

(c) What is the address of *A*[5][6]?

備註	試題隨卷繳交

考　試　科　目	程式設計與資料結構	所　　別	數位內容碩士學位學程/資訊技術組 5152	考 試 時 間	3 月 7 日(日) 第 4 節

5. **(10 %)** Assume that we are given an infix expression: 3+4*(5*6-7).

(a) Draw its expression tree.

(b) Suppose that we are using a stack to convert the expression from the infix to the postfix notation. What is the maximum number of symbols that will appear on the stack at one time during the conversion of this expression?

(c) What is the resulting postfix expression?

6. **(10 %)** Consider inserting the following keys into a hash table of length m=13:

152　44　39　22　134　53　144　131　0　135

The hash function is (k mod m). Please draw the resulting hash table if we use

(a) linear probing; and

(b) quadratic probing function of $F(i)=i^2$, respectively.

7. **(8 %)** The largest element in a heap always appears in position 1, and the next largest element could be in position 2 or position 3. Please give the lists of the smallest (earliest) and the largest (latest) possible positions in a heap of size 15 for the kth largest element, for k=3, 4, 5, respectively. Assume that the values of all elements are distinct.

8. **(14 %)** Consider a set of unsorted numbers in a binary tree. For a given input element x, we will try to find the number that is the closest to x in the tree. That is, y is the answer if $|x-y|$ is the minimum. Write a *recursive* procedure in a language of your choice (please specify) to return the closest number. If the language is in C, the data structure and function prototype are as follows.

```
struct treenode {
  int data;
  struct treenode *left;
  struct treenode *right;
}
int near(int x, struct treenode *tree)
```

備註	試　題　隨　卷　繳　交