

# 銘傳大學九十學年度轉學生招生考試

七月二十九日 第三節

資管 轉三

資料結構 試題

一、 ( 16 points)

- Draw a tree that is both a heap and a binary search tree.
- A binary tree has eight nodes. The inorder and postorder traversal of the tree are given below.  
Draw the tree.  
Postorder: FECHGDBA  
Inorder: FCEABHDG
- Given the following postfix expression, write the minimally parenthesized infix expression:  
 $AB + C - BAD - D/**$
- Draw the corresponding expression tree in part c.

二、 ( 15 points)

- Build a binary search tree, given the keys added in the following order:  
17 18 5 20 27 29 9 33 11 10
- Is this an AVL tree? Why or why not?
- Draw threaded links in place of the null links in this binary tree, where a left null link points to the previous inorder node, and a right null link points to the next inorder node.
- Using this threaded tree, can the threaded links be used to find the parent of each node? If not, give an example; if so, explain how to find the parent with key 29 using the threaded links. The generalized your algorithm.

三、 ( 8 points)

- Draw a Fibonacci tree with 20 nodes and height 5.
- Color this as a red-black tree.

四、 ( 15 points)

- After two passes of a sorting algorithm, the following array:  
47 3 21 32 56 92  
has been rearranged as shown below.  
3 21 47 32 56 92  
Which sorting algorithm is being used?
- In what way(s) is a circular queue better than a linear queue?
- What is the minimum number of keys that can be placed in a B-tree of order 17 and of height 3(four levels)?

五、 ( 10 points)

Insert the keys k: [25, 76, 18, 9, 41, 56] into an initially empty hash table of size 7 using the hash

function  $h1(k) = k \bmod 7$ , giving the total number of collisions for each of the following two cases:

Quadratic collision resolution, using  $i(i+1)/2$  as the total displacement from the initial address for step  $i$ .

Double hashing with  $h2(k) = 1 + [k \bmod 3]$  representing the incremental displacement from the previous address.

六、 ( 18 points)

- (a) How long does it take to insert a new element into a heap? To return the smallest thing in a min-heap? To delete the smallest thing in a min-heaps? To find the largest thing in a min-heap? 8%
- (b) What is the maximum and minimum number of elements in a heap of height  $h$ ? 4%
- (c) Where in a max-heap might the smallest element reside? 3%
- (d) Which of the following sequences are heaps? 3%
  - (i) 42 35 37 20 14 18 7 10
  - (ii) 42 35 18 20 14 30 10
  - (iii) 20 20 20 20 20 20 20

七、 ( 8 points) State whether true or false.

- (a) All AVL-trees are binary search trees.
- (b) Inserting an element in an AVL-tree may require up to  $\log(n)$  rotations, where  $n$  is the number of nodes in the tree.
- (c) Hash tables are attractive because the operations insert and delete have  $O(1)$  worst case running time.
- (d) With AVL-trees we can find the maximum element and the minimum element in  $O(\log n)$  time.

八、 ( 10 points) Multiple choice: only one answer for each question

(1) What is wrong with the following?

```
if sum == SIZE
{
    result = total / SIZE;
    cout << result << endl;
}
```

- (a) cout not allowed in if block
- (b) expression not in parentheses
- (c) expression should have been  $sum = SIZE$
- (d) no semi-colon after

(2) Suppose  $size == MEDIUM$ . What value will  $x$  be after the switch statement?

```
switch (size)
{
    case SMALL;
        x = 100;
    case MEDIUM;
        x = 200;
    case LARGE;
```

```
x += 300;
break;
}
```

(a) 100  
(b) 200  
(c) 300  
(d) 500

- (3) The program containing the if statement below is not working. What is MOST LIKELY wrong?

```
float speed;
float rate;
...
if ( speed == rate )
```

- ...
- (a) Float values should not be tested for equality  
(b) There should be no parentheses  
(c) The == should most likely be just =  
(d) The “if” should most likely be “while”

- (4) What is the difference between lines (2) and (3) below?

```
(1) char name[30];
(2) cin>> name[i];
(3) cin>>name;
```

- (a) (2) and (3) do exactly the same thing.  
(b) (2) reads a character string. (3) reads a single character.  
(c) (2) reads a character string that includes blanks. (3) reads a character string up to but not includes blanks. (3) reads a character string up to but not including a blank.  
(d) (2) reads a single character. (3) reads a character string.

- (5) Which statement below correctly outputs x, y, and z followed by an end-of line?

(a) cout >> x >> y >> z >> endl;  
(b) cout << x, y, z << endl;  
(c) cout << x << y << z;  
(d) cout << x << y << z << endl;

〈 試題完 〉