

Code: AC06 / AT06

Subject: DATA STRUCTURES & ALGORITHM DESIGN

Time: 3 Hours

Max. Marks: 100

DECEMBER 2007

NOTE: There are 9 Questions in all.

- Question 1 is compulsory and carries 20 marks. Answer to Q. 1. must be written in the space provided for it in the answer book supplied and nowhere else.
 - Out of the remaining EIGHT Questions answer any FIVE Questions. Each question carries 16 marks.
 - Any required data not explicitly given, may be suitably assumed and stated.
-

Q.1 Choose the correct or best alternative in the following: (2x10)

a. An infix expression can be converted to a postfix expression using a

- | | |
|-------------|-------------------|
| (A) Stack | (B) Queue |
| (C) Dequeue | (D) None of these |

b. A data structure in which an element is added and removed only from one end, is known as

- | | |
|------------------------|-----------------------|
| (A) Queue | (B) Stack |
| (C) In-built structure | (D) None of the above |

c. If the inorder and preorder traversal of a binary tree are D,B,F,E,G,H,A,C and A,B,D,E,F,G,H,C respectively then the postorder traversal of that tree is

- | | |
|---------------------|---------------------|
| (A) D,F,G,A,B,C,H,E | (B) F,H,D,G,E,B,C,A |
| (C) C,G,H,F,E,D,B,A | (D) D,F,H,G,E,B,C,A |

d. In a binary tree, the number of terminal or leaf nodes is 10. The number of nodes with two children is

- | | |
|--------|--------|
| (A) 9 | (B) 11 |
| (C) 15 | (D) 20 |

e. Which amongst the following cannot be a balance factor of any node of an AVL tree?

- | | |
|-------|--------|
| (A) 1 | (B) 0 |
| (C) 2 | (D) -1 |

f. How many distinct binary search trees can be formed which contains the integers 1, 2, 3?

- | | |
|-------|-------|
| (A) 6 | (B) 5 |
| (C) 4 | (D) 3 |

g. The sort which inserts each elements $A(K)$ into proper position in the previously sorted sub array $A(1), \dots, A(K-1)$

- | | |
|--------------------|-----------------|
| (A) Insertion sort | (B) Radix sort |
| (C) Merge sort | (D) Bubble sort |

h. Direct or random access of elements is not possible in

- | | |
|-----------------|-------------------|
| (A) Linked list | (B) Array |
| (C) String | (D) None of these |

i. Level of any node of a tree is

- | |
|--|
| (A) Height of its left subtree minus height of its right subtree |
| (B) Height of its right subtree minus height of its left subtree |
| (C) Its distance from the root |

(D) None of these

j. A desirable choice for the partitioning element in quick sort is

- (A) First element of the list
- (B) Last element of the list
- (C) Randomly chosen element of the list
- (D) Median of the list

**Answer any FIVE Questions out of EIGHT Questions.
Each question carries 16 marks.**

Q.2 a. Write code to implement a queue using arrays in which insertions and deletions can be made from either end of the structure (such a queue is called deque). Your code should be able to perform the following operations

- (i) Insert element in a deque
- (ii) Remove element from a deque
- (iii) Check if deque is empty
- (iv) Check if it is full
- (v) Initialize it

(12)

b. What are stacks? List 6 different applications of stacks in a computer system. (4)

Q.3 a. Write a recursive code to compute the sum of squares as shown in the series

$$n^2 + (n+1)^2 + \dots + n^2$$

for m, n integers $1 \leq m \leq n$

(8)

b. Give mathematical recursive definition of an AVL tree. (4)

c. Given the following recursive code. Point out the possible errors in the code.

```
//num is a positive integer
int fac(int num)
{
  if (num ≤ 1)
    return 1;
  else {
    return num*fac(num+1);
  }
}
```

(4)

Q.4 a. What are the properties of a *heap*? Distinguish between a *max heap* and a *min heap*. (4)

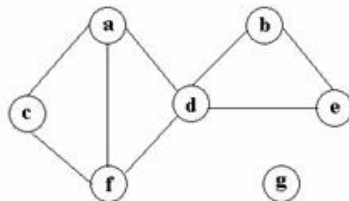
b. Transform the array 2 8 6 1 10 15 3 12 11 into a heap with a bottom up method (8)

c. What are threaded binary trees? Explain inorder threading using an example. (4)

Q.5 a. Write an algorithm to implement Depth-first search? How is Depth-first search different from Breadth-first search? (10)

b. Represent the following graph

- (i) Adjacency list
- (ii) Adjacency matrix
- (iii) Incidence matrix



(6)

Q.6 a. Define B-tree of order m? When is it preferred to use B-trees? (4)

b. Write an algorithm to search a key in a B-tree? What is the worst case of searching in a B-tree? List the possible situations that

can occur while inserting a key in a B-tree? **(8)**

c. What is the complexity of the following code?

```
int counter = 0;
for (i=0; i<n; i++)
    for (j=0; j<n*i; j++)
        counter++; (4)
```

Q.7 a. Show under what order of input, the insertion sort will have worst-case and best-case situations for sorting the set {142, 543, 123, 65, 453, 879, 572, 434}. **(8)**

b. Explain how Merge Sort sorts the following sequence of numbers using a diagram {142, 543, 123, 65, 453, 879, 572, 434}. **(8)**

Q.8 a. Given a stack *s* and a queue *q*, show the contents of each after the indicated operations. The starting contents of *s* and *q* are shown. If an operation would result in an error, write “error” and assume the contents of *s* and *q* do not change. If there is no change otherwise, leave the cell blank or use ditto marks (“”). **(8)**

Operation	Contents of stacks		Contents of queue q	
	Top	bottom	front	rear
Start	empty		2, 4	
s.pop()				
s.push(3)				
q.add(5)				
s.push(q.peek())				
q.add(s.peek())				
q.add(s.pop())				
s.push(s.pop())				
q.add(q.remove())				

b. If *x* is a pointer to a node in a doubly linked list, then *x->prev* is the pointer to the node before it and *x->next* is the pointer to the node after it.

What does this pair of statements, executed in order, do?

```
(x->next)->prev = x->prev;
(x->prev)->next = x->next;
```

If we then execute this pair of statements (after executing the first pair of statements), what happens?

```
(x->next)->prev = x;
(x->prev)->next = x; (4)
```

c. List the following functions in an increasing order of the function order of growth $n!, \sqrt{n}, n^4, n \log n, n, n^2, n^3, \log n, 2^n$ **(4)**

Q.9 a. Consider an insertion of the key = 222 into the hash table shown in the figure. Calculate and indicate the sequence of table entries that would be probed and the final location of the insertion for the key for linear probing and quadratic probing method. The hash function is $h(y) = y \text{ mod } 10$, so for the key = 222, $h(x) = 2 \text{ mod } 10$. In both cases start with the same initial table. Example calculation given at index 2 **(8)**

0		
1		
2	152	$h(222) = 2 \text{ mod } 10$
3	53	
4		
5	75	
6	136	
7	27	
8		
9	999	

b. Using Dijkstra’s method find a spanning tree of the following graph. **(8)**

