ADVANCED DATA STRUCTURES
(CSEN 5101)

Time Allotted : 3 hrs **Full Marks : 70**

Figures out of the right margin indicate full marks.

Candidates are required to answer Group A and any 5 (five) from Group B to E, taking at least one from each group.

Candidates are required to give answer in their own words as far as practicable.

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternative for the following: **10 × 1 = 10**

(i) A binary search tree T is first traversed in inorder and then in postorder. It is found that the sequences in which the nodes are visited are *not* identical in the two cases. Then it must be the case that
(a) at least one node in T has a left child;
(b) at least one node in T has a right child;
(c) at least one node in T has both a left child and a right child;
(d) all nodes in T have two children.

(ii) To sort four integers using insertion sort, the total number of pairwise comparisons in the best case is
(a) 5 (b) 4 (c) 3 (d) 2.

(iii) Consider the following postfix expression with single digit operands, where the operators have their usual meaning:
8 2 3 ^ / 2 3 * + 5 1 * -
Which of the following correctly represents the top two elements of the stack just after the first * is evaluated:
(a) 6, 1 (b) 5, 7 (c) 3, 2 (d) 1, 5.

(iv) A hash table TAB of size m containing n integer entries, where 0 < n < m, has been created using open addressing with linear probing. We now realize that we need to find both the largest and the smallest numbers in TAB. We will be able to do this in time that is
(a) proportional to n (b) proportional to m
(c) independent of both n and m (d) proportional to (n+m).

(v) We want to re-arrange (i.e., permute) the six letters a, a, a, c, c, d, in linear order, but we have to satisfy the condition that the first letter is 'a' and the last letter is 'c'. In how many different ways can we arrange the letters?
(a) 12 (b) 24 (c) 120 (d) 180.

(vi) It is given that a binary tree T has 10 internal (i.e., non-leaf) nodes and 6 leaf nodes. Then the total number of edges in T is
(a) 12 (b) 13 (c) 14 (d) 15.

(vii) An array A contains a sequence of 66 open '('and closed')' parentheses. We have been asked to check whether the sequence of parentheses in A is well-formed (i.e., balanced) using a pushdown stack. To achieve this, we need a stack (ignoring the space for the bottom-of-stack marker) of size at least:
(a) 66 (b) 33 (c) 22 (d) 11.

(viii) A positive integer n when expressed in octal (i.e., radix 8) notation has 12 octal digits (with no leading zeroes). If n is now expressed in hexadecimal (i.e. radix 16) notation, the number of hexadecimal digits will be
(a) 11 (b) 10 (c) 9 (d) 8.

(ix) A table TAB contains the MIN-HEAP (3 5 7 9 11 13 15 17 19 21) of 10 entries. A new element 4 is now placed to the right of 21 in TAB and is then inserted into the heap in the usual manner. Another new entry 6 is now placed in TAB to the right of the 11 entries already there and then inserted into the heap in the usual manner. The resulting MIN-HEAP after these two operations is
(a) (3 4 6 5 9 7 15 17 19 21 11 13) (b) (3 4 6 9 5 7 15 17 19 21 11 13)
(c) (3 4 5 6 7 9 11 13 15 17 19 21) (d) (3 4 6 9 5 7 19 17 15 11 13 21).

(x) How many pairwise comparisons are made when searching for key K = 10 using binary search in the 8-element sorted list (1, 2, 3, 5, 6, 7, 9, 10) stored in the table TAB[0.7] ?
(a) 5 (b) 4 (c) 3 (d) 2.

Group – B

2. (a) Let S be the following set of 16 positive integers: S = { 3, 5, 6, 7, 8, 11, 12, 14, 16, 17, 19, 34, 35, 42, 54, 65 }.

i) Construct a binary search tree (BST) T1 of height = 5 on the above set of keys that has exactly two keys at level 5. (Note that the root is at level 0.)
ii) Determine the average number of comparisons for the successful search of a key in T1.

ADVANCED DATA STRUCTURES
(CSEN 5101)

Time Allotted : 3 hrs **Full Marks : 70**

Figures out of the right margin indicate full marks.

Candidates are required to answer Group A and any 5 (five) from Group B to E, taking at least one from each group.

Candidates are required to give answer in their own words as far as practicable.

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternative for the following: **10 × 1 = 10**

(i) A binary search tree T is first traversed in inorder and then in postorder. It is found that the sequences in which the nodes are visited are *not* identical in the two cases. Then it must be the case that
(a) at least one node in T has a left child;
(b) at least one node in T has a right child;
(c) at least one node in T has both a left child and a right child;
(d) all nodes in T have two children.

(ii) To sort four integers using insertion sort, the total number of pairwise comparisons in the best case is
(a) 5 (b) 4 (c) 3 (d) 2.

(iii) Consider the following postfix expression with single digit operands, where the operators have their usual meaning:
8 2 3 ^ / 2 3 * + 5 1 * -
Which of the following correctly represents the top two elements of the stack just after the first * is evaluated:
(a) 6, 1 (b) 5, 7 (c) 3, 2 (d) 1, 5.

(iv) A hash table TAB of size m containing n integer entries, where 0 < n < m, has been created using open addressing with linear probing. We now realize that we need to find both the largest and the smallest numbers in TAB. We will be able to do this in time that is
(a) proportional to n (b) proportional to m
(c) independent of both n and m (d) proportional to (n+m).

(v) We want to re-arrange (i.e., permute) the six letters a, a, a, c, c, d, in linear order, but we have to satisfy the condition that the first letter is 'a' and the last letter is 'c'. In how many different ways can we arrange the letters?
(a) 12 (b) 24 (c) 120 (d) 180.

(vi) It is given that a binary tree T has 10 internal (i.e., non-leaf) nodes and 6 leaf nodes. Then the total number of edges in T is
(a) 12 (b) 13 (c) 14 (d) 15.

(vii) An array A contains a sequence of 66 open '('and closed')' parentheses. We have been asked to check whether the sequence of parentheses in A is well-formed (i.e., balanced) using a pushdown stack. To achieve this, we need a stack (ignoring the space for the bottom-of-stack marker) of size at least:
(a) 66 (b) 33 (c) 22 (d) 11.

(viii) A positive integer n when expressed in octal (i.e., radix 8) notation has 12 octal digits (with no leading zeroes). If n is now expressed in hexadecimal (i.e. radix 16) notation, the number of hexadecimal digits will be
(a) 11 (b) 10 (c) 9 (d) 8.

(ix) A table TAB contains the MIN-HEAP (3 5 7 9 11 13 15 17 19 21) of 10 entries. A new element 4 is now placed to the right of 21 in TAB and is then inserted into the heap in the usual manner. Another new entry 6 is now placed in TAB to the right of the 11 entries already there and then inserted into the heap in the usual manner. The resulting MIN-HEAP after these two operations is
(a) (3 4 6 5 9 7 15 17 19 21 11 13) (b) (3 4 6 9 5 7 15 17 19 21 11 13)
(c) (3 4 5 6 7 9 11 13 15 17 19 21) (d) (3 4 6 9 5 7 19 17 15 11 13 21).

(x) How many pairwise comparisons are made when searching for key K = 10 using binary search in the 8-element sorted list (1, 2, 3, 5, 6, 7, 9, 10) stored in the table TAB[0.7] ?
(a) 5 (b) 4 (c) 3 (d) 2.

Group – B

2. (a) Let S be the following set of 16 positive integers: S = { 3, 5, 6, 7, 8, 11, 12, 14, 16, 17, 19, 34, 35, 42, 54, 65 }.

i) Construct a binary search tree (BST) T1 of height = 5 on the above set of keys that has exactly two keys at level 5. (Note that the root is at level 0.)
ii) Determine the average number of comparisons for the successful search of a key in T1.

dbaddccdddadcddcadb

Ignoring the symbol '*d*' we get *baccaccab* which reads the same forwards as backwards, so *S* is reversible. But the string *cbdabddbadacdd* is not reversible. Of course, all four characters '*a*', '*b*', '*c*' and '*d*' need not be present in a given string. It can be assumed that no other characters are present in the string.

Given a string *S* of the indicated form as input, your program should output a message stating whether or not the string is reversible.

(4+ 4 + 4) = 12

(b) Construct a BST T2 on the above set S of keys that has the lowest possible value of the average number of comparisons for the successful search of a key among all BSTs built on S. Explain why T2 has this property.

(3+3) + 6 = 12

3. (a) Insert the following 13 keys in the given sequence into a hash table TAB[0..14] of size 15 using the open addressing method with double hashing: The hash functions are h1(k) = k mod 15, h2(k) = k mod 14, and h(j, k) = (h1(k) + j.h2(k)) mod 15 for j = 0, 1, 2 ....

Key sequence: 26, 12, 36, 18, 25, 34, 42, 14, 6, 13, 20, 4, 10
Determine the average number of comparisons for a successful search.

(b) Given an unsorted set S of n > 0 integer entries, we have a choice between *either*: i) sorting the entries in S and then searching for keys using binary search; *or*: ii) creating a hash table T from S and then searching for keys in T.
Under which situations should we opt for option i) and under which other situations should we opt for option? ii) Give reasons and examples.

(4+2) + 6 = 12

Group – C

4. (a) Suppose you are given an infix expression: **((A + B) — C * (D / E)) + F**
Now find its equivalent postfix expression. Show the steps in detail.

(b) S is a sequence of the following three types of characters:
Open and closed parentheses ( ).
Open and closed curly brackets { }.
Open and closed square brackets [ ].

S is *well-formed* when the following conditions all hold. Parentheses, curly brackets and square brackets must be separately balanced. Open-closed parentheses pairs can only enclose open-closed parentheses pairs. Open-closed curly bracket pairs can enclose open-closed parentheses pairs and open-closed curly bracket pairs. Open-closed square bracket pairs can enclose any of the three types of open-closed pairs. So the sequence S = [[{}]{()()}] is well-formed, but the sequence S = ({}) is not well-formed.

Describe in words (or in flowchart form) an algorithm that will determine whether a given input sequence S is well-formed.

7 + 5 = 12

5. (a) Construct an AVL (height-balanced) tree T of height = 5 on the following set S of 14 keys and indicate the balance factor at each node:

S = { 12, 15, 22, 23, 34, 37, 49, 51, 55, 60, 68, 70, 75, 79 }

(b) Does there exist a Red-Black (RB) tree of black-height = 2 on the above set S of keys?
Let T1 be a Red-Black (RB) tree containing n internal nodes (where n > 0), each containing a key. Determine upper and lower bounds on the height of T1 in terms of n.

(3+3) + 6 = 12

Group – D

6. (a) Construct a B-tree T with minimum degree t = 3 in a step by step manner given the following input sequence of 12 keys: 15, 35, 75, 25, 95, 105, 45, 85, 65, 115, 55, 5.

(b) What are the minimum and maximum possible heights of a B-tree having minimum degree t = 3 that contains the above set of 12 keys? Consider all possible orders of the 12 keys as input.
Why are B-trees widely used for storing data in secondary storage but never used as data structures in main memory?

6 + (4 + 2) = 12

7. (a) Consider a skip list containing a sorted list of n key values. Now write a C function to insert a new key value into that skip list. Write another C function that will search a key value supplied by the user.

(b) A database has 100 million records each of size 5 KB. We want to store this database in a B-tree having minimum degree t = 2000. If the page size is 25 MB, what would be the height of T?

(3+3) + (4+2) = 12

Group – E

8. In a particular pattern matching problem, we are required to find occurrences of the pattern P = "pqrqppqr" in the text T = "qppqrqqprpqrqppqrqppqrqr".
   (i) What is the number of character comparisons made by the simple brute-force method when it tries to find all occurrences of pattern P in text T?
   (ii) Determine the prefix function π for pattern P in the Knuth-Morris-Pratt (KMP) algorithm.
   (iii) What is the number of character comparisons made by the KMP algorithm when it tries to find all occurrences of pattern P in text T?

(4+ 4 + 4) = 12

9. Write a program *reversible.c* in C to solve the following problem:
A character array *A* contains a string *S* of *n* characters, where *n* is a positive integer, each entry in A being '*a*', '*b*', '*c*' or '*d*'. *S* is called *reversible* if the following condition holds. When all occurrences of '*d*' in *S* are ignored, the sequence of characters reads the same forwards as backwards. For example, suppose *S* is