Total No. of Questions—**12**] [Total No. of Printed Pages—**7**

[3762]-806

S.E. (IT) (II Sem.) EXAMINATION, 2010

DATA STRUCTURES AND FILES

(2003 COURSE)

Time : Three Hours **Maximum Marks : 100**

N.B. :— (*i*) Answer *three* questions from Section I and *three* questions from Section II.

(*ii*) Answers to the two Sections should be written in separate answer-books.

(*iii*) Neat diagrams must be drawn wherever necessary.

(*iv*) Figures to the right indicate full marks.

(*v*) Assume suitable data, if necessary.

SECTION I

1. (*a*) Write an algorithm for linear search for a list of n numbers. Compare it with binary search with respect to time and space complexity. [6]

(*b*) Define frequency count. [4]

Find the frequency count for given piece of code :

```
for (i = 1; i < = m; i++)
for (j = 1; j < = i; j++)
printf("x");
```

(*c*) Write pseudo-code for merge operation using DLL. [6]

P.T.O.

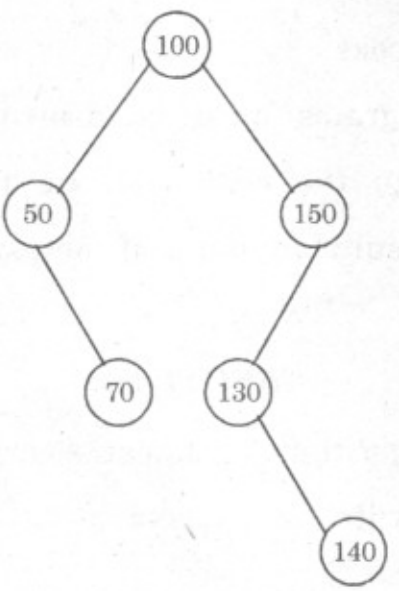
Or

2. (*a*) Write pseudo-code for sparse matrix addition. [6]

(*b*) Write a 'C' function for selection sort. [6]

(*c*) Write a note on asymptotic notations. [4]

3. (*a*) What are the different representations for a binary tree ? Elaborate with formula if necessary. Represent the given binary tree for those representations : [6]

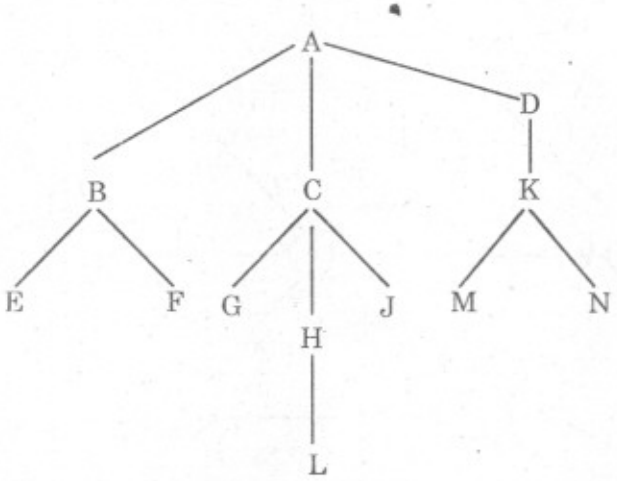


(*b*) Write pseudo-code for inserting into and deleting from a binary search tree. [8]

(*c*) Write a 'C' function for non-recursive in-order traversal. [4]

Or

4. (*a*) List down the steps to convert a general tree to binary tree. Convert the given general tree to binary tree : [6]



(*b*) Write a '*C*' function for non-recursive post-order traversal for threaded binary tree. [6]

(*c*) Write short notes on : [6]

(*i*) Red and Black trees

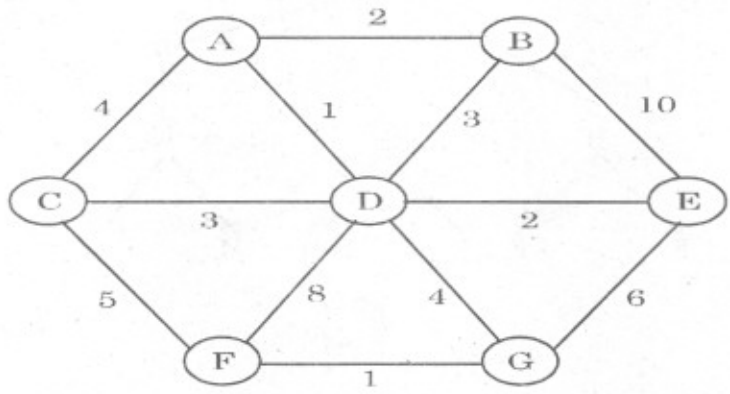
(*ii*) Heap data Structure.

5. (*a*) Define the following with examples : [4]

(*i*) In-degree and out-degree of node.

(*ii*) Adjacent vertices.

(*b*) Write pseudo-code for Prim's method for finding minimum spanning tree for a given graph. Find the MST using Prim's for the given graph : [8]



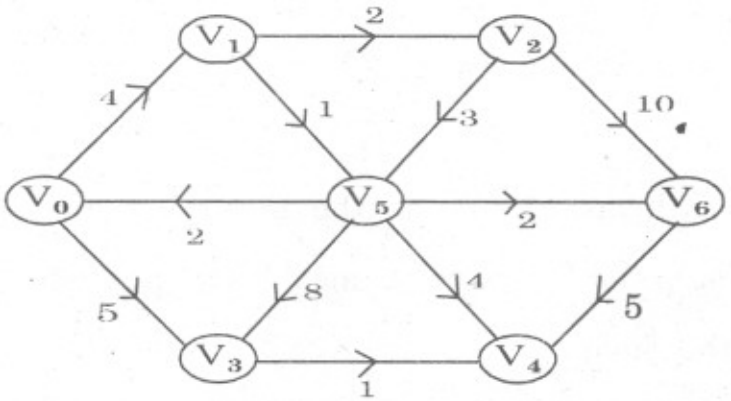
(*c*) Write non-recursive '*C*' function for graph traversal using BFS. [4]

Or

6. (*a*) Write pseudo-code for Kruskal's algorithm for finding minimum spanning tree for any graph. Find the MST using Kruskal's for the graph given in Q. 5(*b*). [8]

(*b*) Write an algorithm to find the shortest path using Dijkstra's algorithm in a graph.

Find the shortest path from V_0 to all other vertices using Dijkstra's algorithm for the given graph : [8]



SECTION II

7. (*a*) Obtain AVL tree for the given data : [10]
MAR, MAY, NOV, AUG, APR, JAN, DEC, JULY, FEB, JUN, OCT, SEP
Show the balance factor and rotation for each step.

(*b*) Write notes on : [8]
(*i*) Symbol table
(*ii*) Static and dynamic trees.

Or

8. (*a*) For the given data build the Huffman's tree and explain each step separately : [9]

Data Item	Frequency
A	15
B	6
C	7
D	12

E	25
F	4
G	6
H	1
J	15

(*b*) Perform heapsort to sort the given list of data items. Show each step for creating a max heap and show each step to sort the heap : [9]

30, 31, 32, 23, 22, 28, 24, 29, 26, 27, 34, 36.

9. (*a*) State the principle of optimality and explain how 0/1 Knapsack problem can be solved using dynamic programming. [10]

(*b*) Write pseudo-code for quicksort. Mention the algorithmic strategy used. [6]

Or

10. (*a*) State job sequencing with deadlines problem and explain how it is solved using greedy strategy with example. [8]

(*b*) Mention the algorithmic strategy for merge sort. Sort the given nos. using merge sort : [8]

40, 60, 70, 50, 20, 10, 30.

11. (*a*) Write pseudo-code for performing primitive operations on sequential file. [8]

(*b*) What is meant by collision ? Explain the following collision avoidance techniques with examples : [8]

(*i*) Linear probing

(*ii*) Chaining without replacement.

Or

12. (*a*) Write a note on hashing and different collision resolution techniques. [8]

(*b*) Write pseudo-code for performing primitive operations on simple index file. [8]