This question paper contains 7 printed pages]

Your Roll No. .....

1976

B.Sc. (Hons.) (Computer Sc.)/III Sem. **C**

Paper 301—ALGORITHMS

(Admissions of 2001 and onwards)

Time : 3 *Hours* *Maximum Marks* : 75

(Write your Roll No. on the top immediately on receipt of this question paper.)

Section A

(*All* the questions from this section are compulsory.)

1. (*a*) Sort the following data using Heap sort : 5

< 25, 57, 48, 37, 12, 92, 86, 33 >

(*b*) Give an efficient in place sorting algorithm to rearrange an array of *n* keys where each key is yellow or blue in colour such that all the yellow keys are arranged before all the blue keys. 5

P.T.O.

(*c*) Show that the second largest of n elements can be found with $n + \lg n - 2$ comparisons in the worst case using tournament method. 5

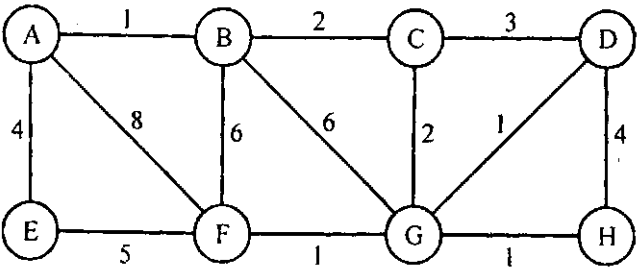
2. (*a*) Create a Binomial Heap that result when the following keys are inserted into an initially empty binomially heap : 5

< 23, 14, 32, 19, 41, 6, 11, 8, 9, 30 >

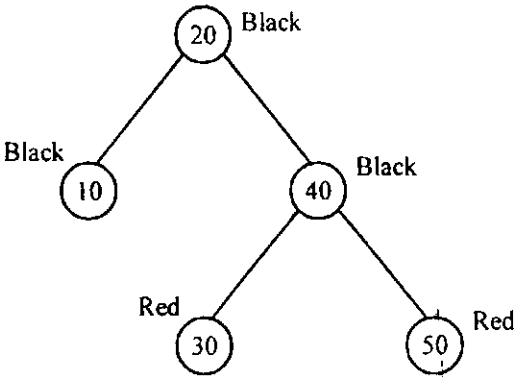
(*b*) Construct an optimal Huffman code for the following set of frequencies : 5

B:5, D:5, A:7, C:8, F:10, E:11

(*c*) Find a minimum spanning tree for a connected undirected graph G as shown using Kruskal's algorithm. Analyze the algorithm for its running time. 5



(*d*) The following RB tree is given : 5



Insert 60, 70 and then delete 40, 60, 10 in given order. Show each tree generated and cases applied explicitly on each step.

Section B

(Attempt any *four* questions.)

3. (*a*) Illustrate the operation of bucket sort on the array :

$A = < .79,.13,.16,.39,.20,.89,.53,.71,.42 >$.

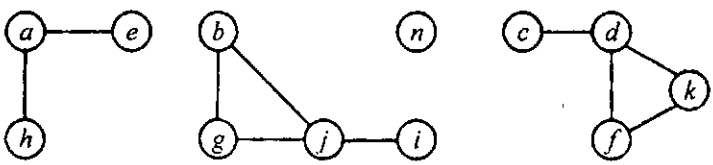
Justify whether it is a stable algorithm or not. 5

(*b*) Given an element 'X' in an *n*-node order statistics tree and a natural number '*i*'. Give an algorithm to find the *i*th successor of 'X' in the linear order of the tree walk in O(lg *n*) time. 5

4. (*a*) Give an optimal parenthesization for the four matrices A(20, 2), B(2, 15), C(15, 40), D(40, 4) using Dynamic Programming. Show the contents of Matrices 'M' and 'S' generated while executing the algorithm. 5

(*b*) A sequence of *n* operations is performed on a Data Structure. The *i*th operation costs *i* if *i* is the exact power of three and 2 otherwise. Use Aggregate Analysis to determine the amortized cost per operation. 5

5. (*a*) Illustrate the execution of connected_components algorithm when run on the following undirected graph G. 5

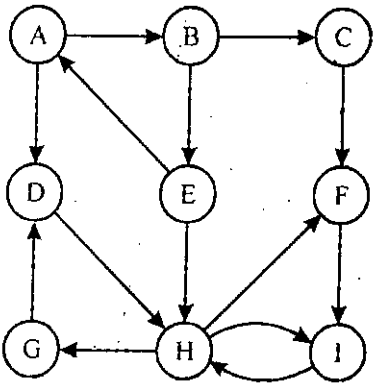


(*b*) Find the Running time of the following Algorithm which tries to find a key in an array a : 5

```
1:    int find(int a[ ], int key) {
2:        int low = 0
3:        int high = a.length – 1;
4:
5:        while (low <= high) {
6:            int mid = low + ((high – low)/2);
7:            int midVal = a[mid];
8:
9:            if (midVal < key)
10:               low = mid + 1
11:           else if (midVal > key)
12:               high = mid – 1;
13:           else
14:               return mid;//key found
15:       }
16:       return –(low + 1);//key not found.
17:       }
```

6. (*a*) Write an algorithm to traverse a graph, represented using adjacency list, using breadth first search. 5

(*b*) Show the order of the vertices produced when Topological sort is executed on the following graph choosing vertices in their alphabetical order. 5



7. (*a*) Show that average running time of quicksort is $O(n \ln n)$. 5

(*b*) Solve the following recurrence relation : 5

$T(n) = T(n - 1) + n$

$T(1) = 1$

8. (*a*) Is it always true that an array that is already sorted is the best case input for a sorting algorithm ? Give an argument or a counter example. 5

(*b*) Can the black-heights of nodes in a red-black tree be maintained as fields in the nodes of the tree without affecting the asymptotic performance of any of the red-black operations ? Show how, or argue why not ? 5