This question paper contains 8 printed pages]

Roll No. | | | | | | | | | | | |

S. No. of Question Paper : **8836**

Unique Paper Code : **234301** **C**

Name of the Paper : **CSHT-305 : Design and Analysis of Algorithms**

Name of the Course : **B.Sc. (H) Computer Sc. Part II**

Semester : **III**

Duration : **3 Hours** Maximum Marks : **75**

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

Question No. **1** of **35** marks is compulsory.

Attempt any *four* questions from Q. No. **2** to Q. No. **7**.

1. (*a*) Give an example of string that would make the brute force string matching algorithm to run in its worst-case (i.e. quadratic) time. 2

(*b*) Consider the following algorithm :

```
recursiveSort(int[ ] List, int n)
 {
    if(n <= 1)
        return list;
   recursiveSort(list, n-1)
    x = list[n];
    for(int i = n-1; i > 0; i++)
```

P.T.O.

```
{
 if( list[i] > x)
   list[i+1] = list[i];
 else
 {
   list[i+1] = x;
   return list;
 }
}
```

Write a recurrence relation for the running time of this recursive version of insertion sort. Justify your answer. 3

(*c*) Analyze the time complexity for the given algorithm (written in pseudo-code) by counting the number of steps. 3

```
x = 0;
y = 1;
for( i = 0; i < n; i++ )
  for( j = 0; j < n; j++ )
  {
    x++;
    y = x * x;
  }
```

(*d*) Given the following procedure for sequential search :

```
int seqSearchRec(int[ ] E, int m, int num, int k)
  int ans;
if(m >= num)
  ans = -1;
else if(E[m] == k)
  ans = m;
else
  ans = seqSearchRec(E,m+1,num,k);
  return ans;
```

Preconditions for seqSearchRec are as follows :

$m >= 0$.

For $m <= i < num$, E[i] is initialized.

Postconditions are as follows :

If ans = -1, then for $m <= i < num$, E[i] != K.

If ans != -1, then $m <= ans < num$ and E[ans]=K.

Prove the correctness of this procedure. 4

(*e*) Which of the following algorithms are inplace :

insertion sort, quick sort, count sort, heapsort. 2

(*f*) Consider the following keys in the order of their appearance :

4, 7, 10.23, 5, 65, 73.

Construct a Red-black tree by doing successive insertions in an initially empty Red-black tree. 5

(*g*) Design an O(|V| + |E|) time algorithm to determine if a given graph G(V, E) is acyclic using BFS/DFS. 4

(*h*) Consider Prim's algorithm to find minimum spanning tree in a graph. Give run time analysis of the algorithm using a suitable data structure, with the help of the table given below (mention the data structure). 5

Operations	**Time taken by the operation**	**No. of times the operation is performed**	**Total time for the operation**
1.			
.			
.			
Total time(of the algorithm)			

(*i*) Consider a k-bit binary counter that counts upward from 0. An array A[0..k-1] is used to store the bits of the counter. A[0] is the lowest order bit and A[k-1] is the highest order bit. One operation can be performed on this counter, INCREMENT. Show that any sequence of n operations performed on this binary counter takes O(n) time. For simplicity, assume that the only operation that takes any time is flipping a bit in the counter. Use aggregate method. 4

(*j*) There are n items in a store. The *i*th item is worth v_i dollars and and w_i pounds of this quantity are available. We wish to fill a knapsack of capacity W pounds. We can pick as much quantity as we want, out of the available, of an item. We wish to maximize the cost of the knapsack. This problem is called fractional knapsack problem. This problem can be solved by following greedy strategy: first compute the value per pound i.e. v_i/w_i for each item. Pick as much as possible, the item with the greatest value per pound. Then pick another one with next greater value per pound and so on, until the knapsack is full. Give an example to show that 0-1 knapsack problem cannot be solved using the greedy strategy as described above. 3

2. (*a*) Given an adjacency-list representation of a directed graph, give a linear time i.e. O(|V| + |E|) time algorithm to compute in-degree of every vertex. Justify the complexity of your algorithm. 4

(*b*) Consider the following set of activities $\{a_1, ... a_n)$. An activity starts at a time s_i and finishes at time f_i. Start and finish times of the activities are given in the following table :

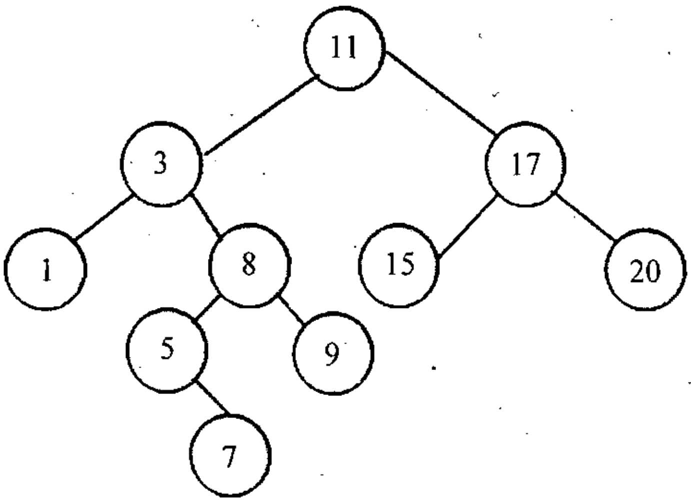
i	1	2	3	4	5	6	7	8	9	10
s_i	1	3	5	5	3	6	8	9	7	10
f_i	4	5	7	10	10	15	16	18	20	24

Find largest subset of mutually compatible activities. 3

(*c*) Construct an example with only three or four matrices where the worst multiplication order does at least 100 times as many elementwise multiplications as the best order. 3

3. (*a*) Analyse Heapsort algorithm giving the running time of max-heapify, build_max_heap and heapsort procedures. 5

(*b*) Color the nodes of the following binary search tree with red or black colors so that the resulting tree is a Red-Black tree.Delete 17, 7, 8 sucessively from the tree obtained in the previous steps. 2+3

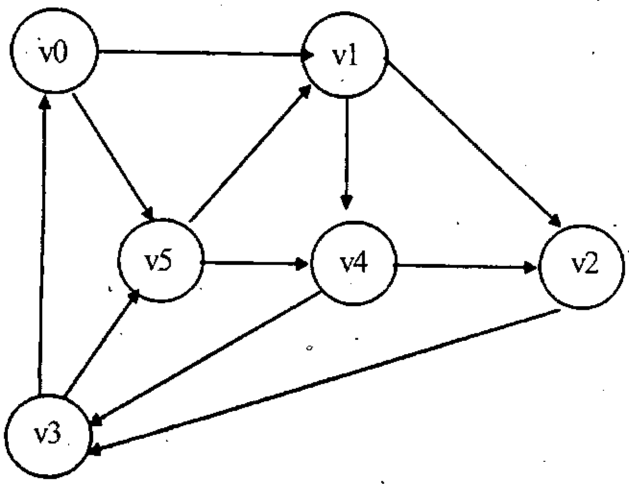


4. (*a*) . Consider the following English words (taken from the alphabet (A..J)) :

BED, BAD, HID, BID, DIG, BAG, CAB, AID.

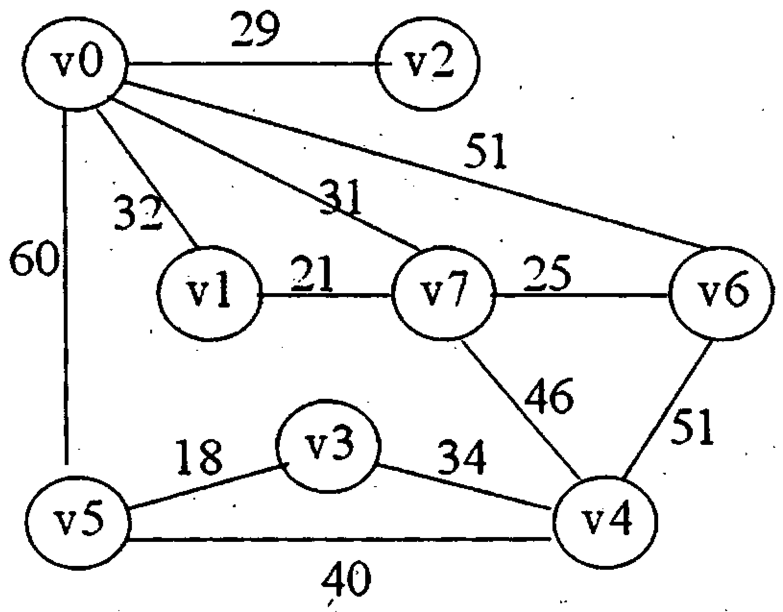
Illustrate the operation of sorting this list in alphabetical order using Radix sort. 3

(*b*) Given below is a directed graph. Give Depth first search forest which you obtain by applying DFS algorithm starting with vertex v5. Classify the edges of the graph as back edges, forward edges, tree edges and cross edges. 4



(*c*) Write an algorithm to sort 4 keys in 5 comparisons. 3

5. (*a*) Compute the minimum spanning tree using Kruskal's algorithm. Show the final spanning tree generated by algorithm. 5



(*b*) Compute the fail-indexes used by KMP algorithm for the following pattern : P: aabcaaabacac. 5

6. (*a*) Give an example of a directed graph such that the resultant DFS forest may have more than one connected component. Your graph should have at least 6 vertices and 10 edges. Specify the vertex at which depth first search should begin so that resultant forest is as desired. 3

(*b*) Suppose we use randomized select to select the minimum element of the array A = <3, 2, 9, 0, 7, 5, 4, 8>. Describe a sequence of partitions that results in a worst-case performance of Randomized-Select. 3

(*c*) We define the Fibonacci numbers as follows :

F(0) =0

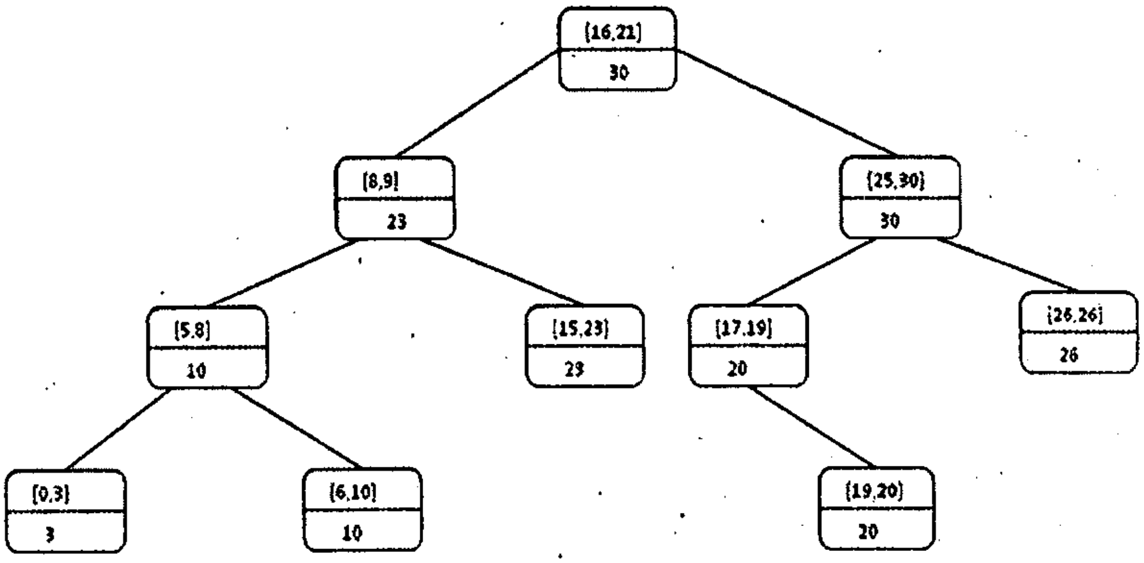
F(1) =1

for all n > 1,Fn = F(n-1) + F(n-2)

Write fib_memoize(n). It should be recursive but it should memoize its answers, so that it runs in time O(n). 4

P.T.O.

7. (*a*) Find out whether their exists an interval x in the interval tree given below which overlaps with interval [18, 24]. Show the sequence of nodes compared in the process. 4



(*b*) Consider stack operations Push, Pop and Multipop, that can be performed on a stack data structure. Use accounting method to show that any sequence of n operations performed on an initially empty stack takes O(n) time. 3

(*c*) What are the two key factors that decide whether dynamic programming is applicable for an optimization problem or not ? 3