53 (CS 503) DAAL

2017

DESIGN AND ANALYSIS OF ALGORITHM

Paper : CS 503

Full Marks : 100

Time : Three hours

The figures in the margin indicate full marks for the questions.

*Answer **any five** questions.*

1. *(a)* Consider the following function

```
int fun1 (int low, int high, int key)
{
  while (low <= high)
  {
    int mid = (low+high)/2 ;
    if (a[mid] < key)
```

```
        {
            low = mid + 1 ;
        }
        else if (a[mid] > key)
        {
            high = mid-1 ;
        }
        else
        {
        return mid ;
        }
    }
    return -1 ;
}
```

Compute the complexity of the above algorithm. 10

(b) If $f(n)=O(g(n))$ and $g(n)=O(l(n)$ then prove

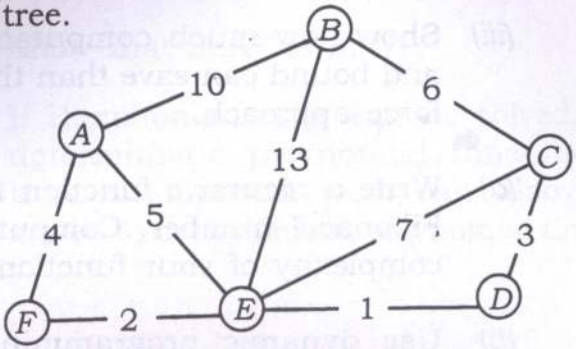
(i) $f(n)=O(l(n))$

(ii) $f(n)+g(n)=O(l(n))$

10

2. (a) Write an algorithm (greedy) to find out the minimum spanning tree of a graph G. Compute the complexity of your algorithm. 5+5

(b) Apply your algorithm on the following graph to compute minimum spanning tree. 5



(c) Consider the following recurrence relation and compute the time complexity, in terms of (H)

$$T(n)=T(n/2)+T(n/2)+n$$ 5

3. Consider the following 0/1 knapsack problem. The size of the knapsack is 7 *kg*. Objects and their profits are like below —

	O_1	O_2	O_3	O_4	O_5	O_6
Size	3	4	5	7	2	6
Profit	6	12	25	14	8	6

(i) Use brute force method to compute the maximum profit.

(ii) Use branch and bound to compute the maximum profit.

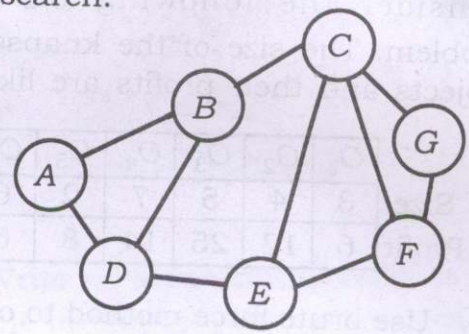
(iii) Show how much computation branch and bound can save than that of brute force approach.

5+10+5

4. (a) Write a recursive function to compute Fibonacci number. Compute the time complexity of your function. 5+5

(b) Use dynamic programming to write another function for computing Fibonacci number. Compute and compare the time complexity with the previous approach. 5+5

5. (a) Consider the following graph and perform breadth first and depth first search. 5+5



(b) Prove that 2SAT is not NP complete. 10

6. (a) What do you mean by the term "reduction"? Discuss with an example. 5

(b) Show that 3SAT is in NP. 5

(c) If Hamiltonian cycle can be solved in deterministic polynomial time-show that Hamiltonian path can also be solved in deterministic polynomial time. 10
