ws Oct 09-1 183

Con. 5681–09.   (REVISED COURSE)   SP–7808

(3 Hours)   [Total Marks : 100

Analysis of Algorithm & Design

N.B. : (1) Question No. **1** is **compulsory**.
  (2) Attempt any **four** questions out of the remaining **six** questions.
  (3) Assumptions made should be **clearly** stated.
  (4) Assume **suitable** data whenever **required**.

1. (a) Consider the iterative program to find fibonacci numbers.   5
Unsigned int fibonacci (Unsigned int n)

```
{
    int previous = –1
    int result = 1
    for (Unsigned int i20; i < = n; ++i)
    {
        int Const Sum = result + previous,
        previous = result;
        result = sum;
    }
    return = Sum;
}
```

Determine the running time of the above algorithm.

 (b) Discuss the best case and worst case efficiency for a quick sort algorithm.   5
 (c) Write and explain with illustration the recursive algorithm for a merge sort.   5
 (d) Explain the Strassen's Matrix Multiplication.   5

2. (a) Explain the following algorithms with an example and give its complexity :   10
   (i) Selection Sort   (ii) Heap Sort.
 (b) Write the algorithm for pattern matching using Boyer–Moore Algorithm. Suppose   10
we have a string – abacaabaccabacabaabb. Illustrate the Boyer–Moore algorithm
to search a string → abacab.

3. (a) What are trics ? What are its applications ? Briefly, describe the types of tries.   10
 (b) Illustrate the heap sort algorithm for the array   10
A = <27, 17, 3, 16, 13, 10, 1, 5, 7, 12, 4, 8, 9>

4. (a) Find the minimum cost spanning tree for the graph using Prim's and Kruskal's   10
algorithm

[TURN OVER
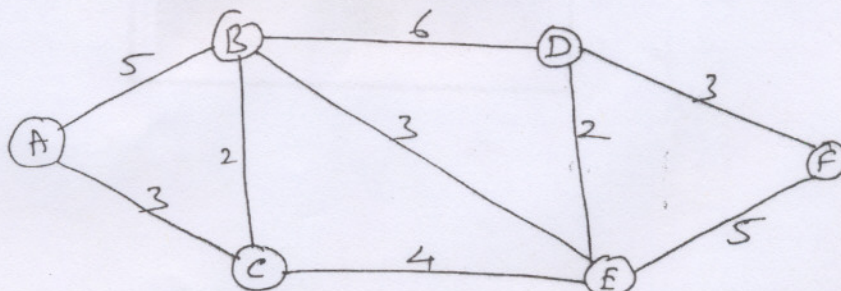
   (b) Write an algorithm for 0/1 Knapsack problem using dynamic programming. **10**
      When do we use the greedy algorithm for the Knapsack problem ? Explain.

5. (a) Find the Huffman code for the following set of frequencies, based on the **10**
      first 8 fibonacci numbers.
          $a = 1, b = 1, c = 2, d = 3, e = 5, f = 8, g = 13, h = 21$.

   (b) Explain the graph coloring algorithm with an example. **10**

6. (a) Explain the 8-queen problem. Write an algorithm using back-tracking to solve **10**
      this problem.

   (b) Compare the Radix sort and Bucket sort algorithm and state its complexity. **10**
      (Write both the algorithms).

7. (a) Explain the Quick Sort algorithm using the Divide and Conquer method. **10**
      Determine the complexity for the worst case/best case and the average case.

   (b) Write short notes on the following : **10**
      (i) Warshall's Algorithm   (ii) Djkstra's Algorithm.