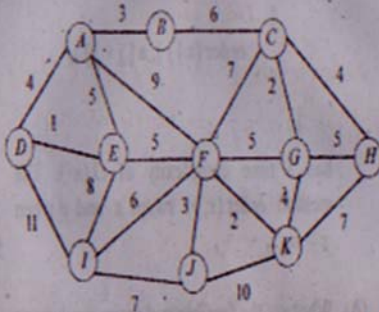


VEER SURENDRA SAI UNIVERSITY OF TECHNOLOGY, BURLA
 DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
 & INFORMATION TECHNOLOGY
 SESSION 2014-2015 (EVEN SEMESTER)

(6)

- (b) Find a minimum spanning tree for a given graph using Kruskal's algorithms? What is the time complexity of this algorithms? Explain the reason why Kruskal's algorithm always yields optimal spanning tree. 5



8. (a) What are the space requirements and complexity (work done) in the all pairs shortest paths problem? 2
- (b) How does the dynamic programming approach differ from the greedy approach? 4
- (c) Show that the greedy approach to the shortest paths problem does not work if the edge costs can be negative. 4

Total Pages-6

(Set-1)

B.Tech-6th
Analysis and Design Algorithm

Full Marks : 70

Time : 3 hours

Answer any six questions including Q. No. 1

The figures in the right-hand margin indicate marks

1. Answer all the following questions: 2 x 10
- (a) Show that 2^{n+1} is $O(2^n)$.
- (b) Define fractional Knapsack Problem. What is the time complexity of the algorithm to solve fractional knapsack problem using greedy paradigm?
- (c) Explain why we expect the average case for merge sort to be almost the same as the worst case.
- (d) How many comparisons does heap sort do in the worst case on an array with four keys? Consider both phases of heap sort.

(2)

- (1-10)
- (e) What is the maximum length of a code word possible in Huffman encoding of an alphabet of n characters?
 - (f) What is the basic objective of Robin Karp pattern matching algorithm?
 - (g) Why do we need approximation algorithm?
 - (h) In Kruskal's method for finding a minimum-spanning tree, how does the algorithm know when the addition of an edge will generate a cycle?
 - (i) Give three properties of NP-complete problems.
 - (j) Which function grows at the faster rate e^n or 2^n ? Justify your answer.
2. (a) Write down the recurrence relation for the work done by the Fast Fourier Transform and explain the origin of the terms. 5

(3)

- (b) Derive an approximate recurrence relation for the average work done by Mergesort to sort an array with n elements. Carefully explain how it was obtained then solve it to obtain a closed expression for the average work done by Mergesort? 5
3. (a) Describe a dynamic-programming algorithm to find the maximum-length common subsequence X of A and B . (Please clearly define your notation). Analyze the running time and space requirements of your algorithms (in terms of big- O notation). 5
- (b) What is the principle of optimality that is used in dynamic programming paradigm? Explain with an example how use of table is found to be efficient instead of using recursion. Write an algorithm to find the minimum cost path from source to sink in a multistage graph using backward approach. What is the time complexity of the algorithm? 5

4. (a) Construct a Huffman code for the following data

Character	A	B	C	D	-
Probability	0.4	0.1	0.2	0.15	0.15

Encode the text *ABACABAD* using the Huffman code obtained. Decode the text whose encoding is 100010111001010 using the above coding scheme. 5

- (b) Give the algorithm of Binary search. Explain how it functions? Show that the algorithm is of same order for both average and worst case i.e. $O(\log_2 n)$. 5
5. (a) Explain the process of Heap sort. Write an algorithm to construct a min heap? Construct max-heap from the following list. Sketch the heap after deleting 77 and 75. What is the time complexity of the whole process? (21, 64, 56, 63, 44, 7, 9, 77, 75, 32, 34, 14, 49). 5
- (b) Briefly discuss the three different methods available for providing lower bounds of an algorithm. 5

6. (a) Show that the algorithm to sort n numbers:

```
for (int i = a.length-2; i >= 0; i--)
{
    for (int j = 0; j <= i; j++)
```

```
    {
        order(a[j], a[j+1]);
    }
}
```

has a time complexity of $O(n^2)$. The method *order(x, y)* swaps x and y when $x > y$. 5

- (b) What are the feasible and optimal solutions for the minimum spanning tree problem? Which process determines the amount of work done by Kruskal's algorithm? 5
7. (a) Define and differentiate between *P*, *NP* and *NP*-complete problems with examples. Explain why input size is an important consideration for algorithm complexity? 5