C4-R4: ADVANCED ALGORITHMS

NOTE:

1.	**Answer question 1 and any FOUR from questions 2 to 7.**
2.	**Parts of the same question should be answered together and in the same sequence.**

Time: 3 Hours **Total Marks: 100**

1.

a) Define Recurrence relations with an example and its solution steps.
b) Define NP-complete problems.
c) What is the main difference between algorithm and program?
d) What output one might get if Binary search is applied on a list, which is not sorted a priori?
e) Define "complexity" term in the context of computer algorithms.
f) For 0/1 Knapsack problem if $c[i, w]$ to be the solution for items 1, 2, …, i and maximum weight is w, then define $c[i, w]$.
g) What is an approximate algorithm? Explain.

(7x4)

2.

a) Discuss Boyer-Moore algorithm for string matching.
b) Discuss the role of string matching in sequence alignment and its applicability for two sequence alignment and multiple sequence alignment.

(8+10)

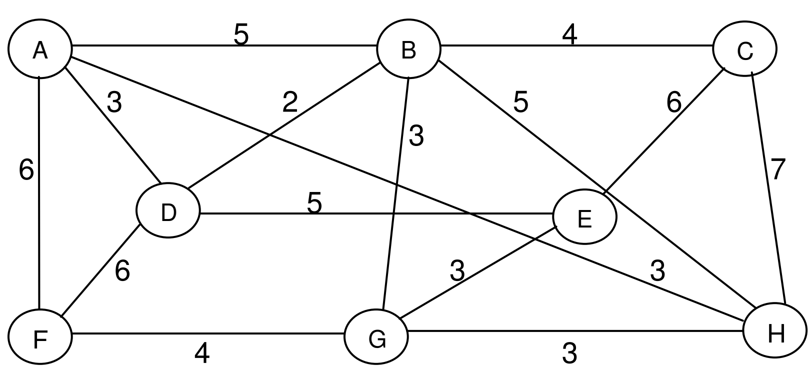
3.

a) Show the average case complexity analysis of binary search.
b) Perform worst case time complexity analysis of quick sort.

(10+8)

4.

a) The English coinage before decimalization included half-crowns (30 pence), florins (24 pence), shillings (12 pence), sixpences (6 pence), threepences (3 pence), and pennies. Show that with these coins the greedy algorithm (already designed or you may design) does not necessarily produce an optimal solution, even when an unlimited supply of coins of each denomination is available.
b) For the following graph, find minimum spanning tree using any known algorithm.



(8+10)

5.

a) What is dynamic programming? Why it is called so? Solve the knapsack problem of five objects, whose weights are respectively 1, 2, 4, 5 and 7 units, and whose values are 1, 4, 6, 12 and 16. The maximum of 8 units of weight can be carried.

b) Solve the following assignment problem efficiently.

	1	2	3	4
a	11	18	13	28
b	16	23	11	21
c	13	19	13	15
d	11	15	23	22

(9+9)

6.

a) Discuss about the computational complexities of algorithm and hence the classification of algorithms into P and NP classes.

b) What is amortized analysis? Discuss polential method using example.

c) If P is the set of all decision problems those are solvable in polynomial time, then show that P = {L: L is accepted by a polynomial-time algorithm}.

(4+9+5)

7.

a) Discuss vertex cover problem.

b) State and prove the Euler theorem.

c) Explain Radix sort.

(6+6+6)