C4-R4 :ADVANCED ALGORITHMS

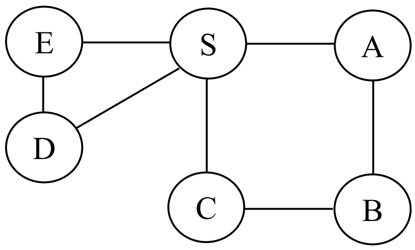
NOTE :

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) Solve T(n) = T(n/4) + T(n/2) + cn^2 ; T(1) = c; T(0) = 0, where c is a positive constant.

(b) What is the basic intuition behind breadth-first-search ? Give the result of BFS on the following graph :



(c) A long string consists of the four characters A, C, G, T; they appear with frequency 31%, 20%, 9% and 40% respectively. What is the Huffman encoding of these four characters ?

(d) The Fibonacci numbers, F_n are generated by the simple rules

$$F_n = \begin{cases} F_{n-1} + F_{n-2} & if\ n > 1 \\ 1 & if\ n < 1 \\ 0 & if\ n = 0 \end{cases}$$

How does the Fibonacci number grow ? If T(n) is the number of computer steps needed to compute Fn, write the recurrence equation for Fibonacci number using it. How T(n) is related to Fn ?

(e) Describe the main ideas behind greedy algorithms. Prove that Greedy Algorithm does not always give optimal solution.

(f) Write the recurrence and time complexity for worst case for Quick soft ?

(g) What do you mean by reduction to second problems ? Give a definition of an NP-Complete problem. When can we have P=NP ?

(7×4)

2. (a) Prove (by using the definitions of the notations involved) or disprove (by giving a specific counter example) the following assertions :

(i) If $t(n) \in O(g(n))$, then $g(n) \in \Omega(t(n))$.

(ii) $\Theta(\alpha g(n)) = \Theta(g(n))$, where $\alpha > 0$.

(iii) $\Theta(g(n)) = \Theta(g(n)) \cap \Omega(g(n))$.

(iv) For any two nonnegative functions t(n) and g(n) defined on the set of nonnegative integers, either $t(n) \in O(g(n))$, or $t(n) \in \Omega(g(n))$, or both.

(b) (i) What are the minimum and maximum numbers of elements in a heap of height *h* ?

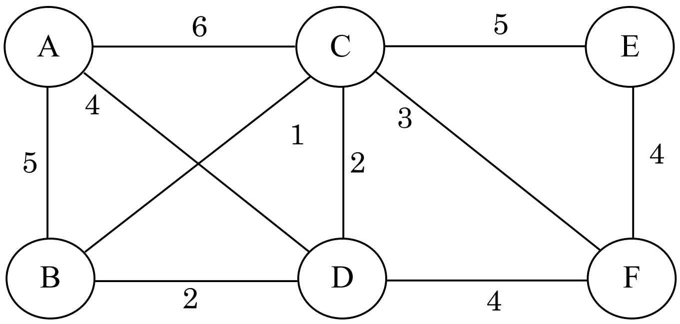
(ii) Illustrate the operation of MAX-HEAP-INSERT on the heap

A = 15,13,9,5,12,8,7,4,0,6,2,1.

(9+9)

3. (a) A sequence of *n* operations is performed on a data structure. The i^{th} operation cost C*i* if *i* is an exact power of 2, and 1 otherwise. Use aggregate analysis to determine the amortized cost per operation.

(b) Give the Kruskal algorithm for the MST. Find the MST using it for the undirected graph given below :



(c) Consider the definition : h(0) = 1; h(n) = h(n – 1) + h(floor(n/2)) + 1. For example, h(2)=h(1)+1=3, h(3)=h(2)+h(1)+1=5, etc. An obvious algorithm to compute h is a recursive one based directly equation. Is that algorithm efficient.Why or why not ? If not, how can the algorithm be improved ? Justify your answer.

(6+6+6)

4. (a) Sort the list E, X, A, M, P, L, E in alphabetical order by selection sort. Is selection sort stable ? Is it possible to implement selection sort for linked lists with the same Θ(n^2) efficiency as the array version ?

(b) Compare Linear and Binary search. Write Iterative and Recursive functions of Binary search.

(9+9)

5. (a) Suppose we want to make change for n cents, using least number of coins of denominations 1, 10, and 25 cents. Describe an O(n)-dynamic programming algorithm to find optimal solution.

(b) Find an optimal parenthesization of a matrix-chain product whose sequence of dimensions is <5, 10, 12, 5, 50, 6>.

(9+9)

6. (a) How many comparison (successful and unsuccessful) are made by the brute-force string-matching algorithm in searching for the following patterns in the binary text of 1000 zeros ?

(i) 00001;

(ii) 10000;

(iii) 01010

(b) The Partition Problem is as follows. You are given a list of positive integers. You would like to create two sublists of the input list, with each number in exactly one of the sublists, so that the sum of the numbers in each sublist is the same. For example, if the input list is 2,2,3,4,5,7,9, then you can partition this into (2,2,5,7) and (3,4,9), each with sum 16. Notice that each number in the input list is present in exactly one of the two sublists. If a number is repeated, it must occur the same number of times total in the sublists.

(i) Express the partition problem as a set.

(ii) Describe a nondeterministic polynomial time algorithm for the partition problem

(9+9)

7. (a) Design an algorithm for the following problem : Given a set of n points in the Cartesian plane, determine whether all of them lie on the same circumference.

(b) (i) Find gcd(31415, 14142) by applying Euclid's algorithm.

(ii) Estimate how many times faster it will be to find gcd(31415, 14142) by Euclid's algorithm compared with the algorithm based on checking consecutive integers from min{m, n} down to gcd(m, n).

(9+9)