H Roll No. 2194026.........

TCS–409

B. TECH. (CSE) (FOURTH SEMESTER) END SEMESTER EXAMINATION, June, 2023

DESIGN AND ANALYSIS OF ALGORITHMS

Time : Three Hours

Maximum Marks : 100

Note : (i) All questions are compulsory.

(ii) Answer any ***two*** sub-questions among (a), (b) and (c) in each main question.

(iii) Total marks in each main question are **twenty**.

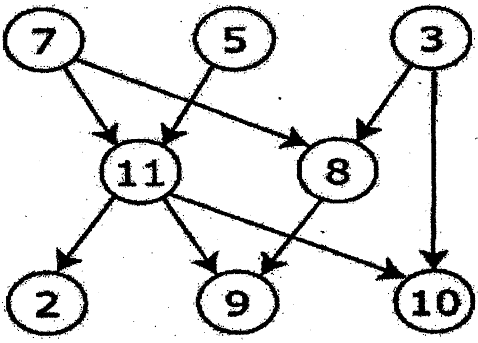
(iv) Each sub-question carries 10 marks.

(v) Write code in C, C++, Java or Python only.

1. (a) What is merge sort ? Explain with code. Use merge sort on the following numbers and show how this divide and conquer algorithm works. (CO1)

Numbers : 9, 11, 4, 2, 3, 6, 8, 12

(b) What is the difference between DFS and BFS. Write the code for topological sort and apply on the following graph : (CO3)



(c) Match each algorithm below with the tightest asymptotic upper bound for its worst-case running time by inserting one of the letters A, B, ..., G into the corresponding box. For sorting algorithms, n is the number of input elements. For matrix algorithms, the input matrix has size n × n. For graph algorithms, the number of vertices is n, and the number of edges is θ (n).

You need not justify your answers. Some running times may be used multiple times or not at all. (CO2)

□	Bubble sort	A : O (lg n)
□	Heap sort	B : $O(n^2)$
□	BUILD-HEAP	C : O (n lg n)
□	Bellman-Ford	D : O (n)
□	Prim's	E : O (E + V)
□	Depth-first search	F : $O(n^{2.5})$
□	Floyd-Warshall	G : $O(n^3)$

2. (a) What is Rabin Karp Algorithm ? How does it work ? Explain with an example. (CO5)

(b) Complete the following using bottom up dynamic programming for Longest common subsequence.

int LCS(string s1, string s2, int m, int n);

where sl and s2 are input strings, and, m and n are their lengths respectively.

Fill the table created using dynamic programming for the following strings. (CO4)

S 1 = "aabcdacdbb"

S2 = "abcdbcdabc"

(c) Complete the following function for calculating min operation for matrix chain multiplication using dynamic programming : (CO4)

int matrixChainMultiplication (int *p, int i, j);

p is the array having the information about the matrices dimensions initially i is 1 and j is n – 1, where n is the number of matrix.

3. (a) A file contains the following characters and their corresponding frequencies as shown below :

a : 36, b : 24, c : 19, d : 18, e : 15, f : 23

We use Huffman coding for data comparisons, generate the encoding for a, b, c, d, e, f using Huffman encoding and find the average length of a character after compression. (CO5)

(b) Solve each of the following recurrences for tight asymptotic upper bound

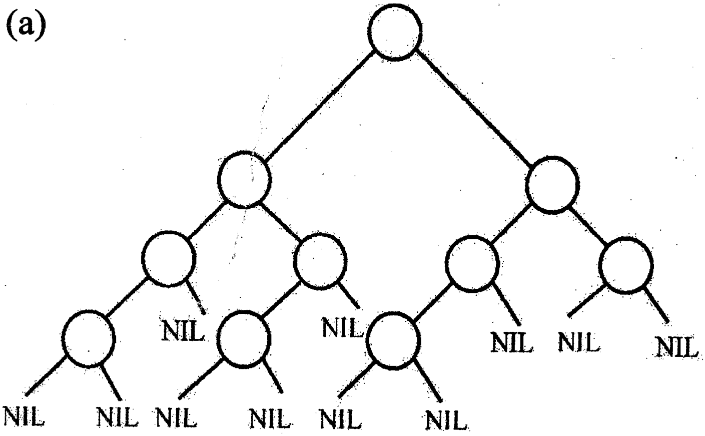
(O notation). Take base case as T(1) = 1, until specified otherwise. (CO1)

(i) $T(n) = T\left(\frac{n}{8}\right) + \sqrt[3]{n}$

(ii) $T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{n}{4}\right) + 5n$

(c) Write the algorithm for insertion sort and explain why insertion sort is better than merge or quick sort in case of almost sorted array. (CO2)

4. (a)



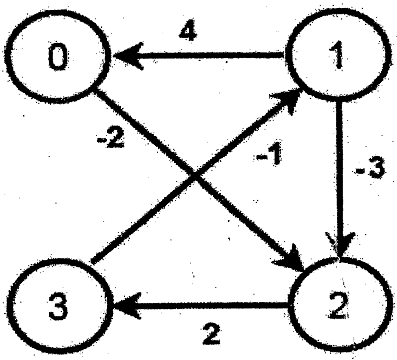
Fill the above tree for BST using the following keys :

17, 87, 19, 13, 14, 44, 56, 80, 90, 101

ss(CO2)

(b) What is hashing ? Define collision and different collision handling techniques. (CO5)

(c) What is the use of Bellman-Ford algorithm ? Write the function for the same. Can we apply Bellman Ford on negative weight edge. If yes, apply on the following graph : (CO4)



5. (a) Write the pseudo code of bubble sort. What will be the output after 5 iterations of bubble sort on the following numbers ?

5, 7, 1, 2, 3, 9, 0, 4, 6 (CO1)

(b) Differentiate between P, NP, NP-Complete, NP- Hard. Explain travelling salesman problem and its approximate solution. (CO4)

(c) Differentiate between greedy and dynamic programming. Write code for fractional knapsack and 0/1 knapsack. (CO3)