JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TEST -3 EXAMINATION- 2024

B.Tech-IV Semester (CSE)

COURSE CODE (CREDITS): 18B11CI412 (03) MAX. MARKS: 35

COURSE NAME: Design and Analysis of Algorithms

COURSE INSTRUCTORS: AKJ, ARV*, DHA, KPR MAX. TIME: 2 Hours

Note: *(a) All questions are compulsory.*

(b)Marks are indicated against each question in square brackets.

(c) The candidate is allowed to make Suitable numeric assumptions wherever required for solving problems

1. Answer the following Questions: **[CO-3, 4][2+2+2+1+2]**

(a) The transpose of a directed graph $G = (V, E)$ is the graph $G^T = (V, E^T)$, where $E^T = \{(v, u) \in V \times V : (u, v) \in E\}$. That is, G^T is G with all its edges reversed. Describe efficient algorithms for computing G^T from G, for both the adjacency-list and adjacency-matrix representations of G. Analyze the running times of your algorithms.

(b) Show the d (distance in term of edges from the source) and π (predecessor in the breadth-first tree) values that result from running breadth-first search on the directed graph as shown in Fig. 1, using vertex 3 as the source.

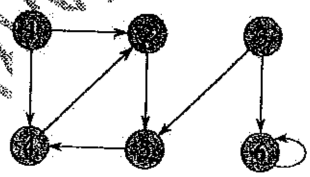

Fig. 1: A directed Graph G = (V, E)

(c) Identify the tree edges, back edges, forward edges, and cross-edges resulting from running the DFS algorithmon the graph shown in Fig. 2.

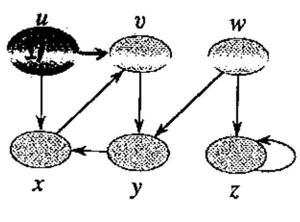

Fig. 2: A directed Graph G = (V, E)

(d) What conditions must the input graph satisfy to perform a topological sort?

(e) Provide the ordering after running the topological sort on the graph in Fig. 3.

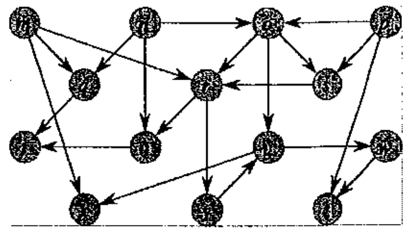

Fig. 3: A directed Graph G = (V, E)

2. Answer the following Questions: **[CO-3][4+5]**

(a) Given a directed weighted graph in Fig. 4, implement a greedy based algorithm to find the shortest path from a given source **"a"** to all other vertices in the graph. But the twist is that the weight of undiscovered edges increase by a step size of *4* each time an edge with the least weight among the discovered edges is added into the shortest path sequence.

(b) In the below graph, if all edge weights have the same value, then give pseudo code of the most optimal algorithm that would give optimal results. Trace the algorithm step wise to compute the shortest distance from source "a" to all vertices. Also, give time and space complexity of the algorithm used.

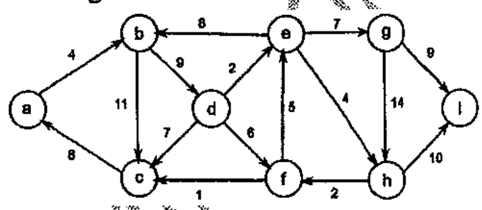

Fig. 4: A directed Graph G = (V, E)

3. Consider the Graph G given in Fig. 4 as undirected graph. Now, we want to find out the optimal way to connect (in terms of connection cost) all the given vertices of the graph, and then answer the following questions: **[CO-4][3+3+3]**

1. What is the structure of the generated output?
2. Step-by-step solution and mention the optimal cost.
3. Analyze the complexity of the algorithm that you used to generate the output along with the data structures that you used in the algorithm.

4. (a) Given a pattern **"AABAACAABAA"** and a text **"AABAACAADAABAABA"**, after how many character comparisons will the KMP algorithm find the first occurrence of the pattern in the text? Explain the process. Provide an example where the KMP algorithm significantly outperforms the naive string matching algorithm.

(b) Given the text **"3141592653589793"** and the pattern **"26535"**, use the Rabin-Karp algorithm to find the starting index of the first occurrence of the pattern in the text. Assume the base d is *10* and the prime number q is *13*. Provide an example where the Rabin-Karp algorithm might perform poorly compared to other string matching algorithms like KMP.

(c) What do you mean by reducibility in context of NP- Hard problems? **[CO-5][3+3+2]**