

# **SNS COLLEGE OF ENGINEERING**

Kurumbapalayam (Po), Coimbatore - 641 107



## AN AUTONOMOUS INSTITUTION

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

**UNIT IV-2 MARKS** 

Fourth Semester B.E. Computer Science and Technology 23TSB202 – Analysis of Algorithm Regulations 2023

#### **1.What is a Flow Network?**

**Ans:** A Flow network is a directed graph where each edge has a capacity that represents the maximum amount of flow it can carry. The graph includes a source node where flow originates and a sink node where flow terminates. The flow on each edge must respect the capacity constraints and flow conservation rules.

2.What is the Ford-Fulkerson method, and what is it used for in network analysis? Ans: The Ford-Fulkerson method is an algorithm used to compute the maximum flow in a flow network from a source to a sink. It works by repeatedly finding augmenting paths and increasing the flow until no more such paths exist. The process continues until no more augmenting paths exist in the residual graph.

In network analysis, it helps optimize resource allocation like traffic, data, or fluid flow.

#### 3. Define Augmentation Path.

**Ans:** The algorithm works by searching for paths from the source to the sink where additional flow can be pushed. These paths are called augmenting paths, and the flow is updated accordingly in each iteration. The method repeats until no such path can be found.

#### 4. Interpret Residual Graph.

**Ans:** A residual graph shows how much additional flow can be sent through each edge. It is constructed based on the current flow and capacities. If an edge is fully used, it may even show a reverse edge with flow value to allow for backtracking.

### 5. What is Bipartite Graph?

**Ans:** A bipartite graph is a graph whose vertices can be divided into two distinct sets such that no two vertices in the same set are connected. All edges only connect a vertex from one set to a vertex in the other set. This property makes them useful in matching problems.

#### 6.Inspect Matching in Bipartite Graph.

Ans: A matching is a set of edges such that no two edges share the same vertex. In a bipartite graph, this

involves pairing vertices from one set with those in the other. The goal is often to find the largest possible matching, known as maximum matching.

**7.How can maximum matching in bipartite graphs be solved using flow networks? Ans:** We can transform the bipartite graph into a flow network by adding a source and sink node. Each edge is given a capacity of 1, and then the Ford-Fulkerson method is applied. The value of the maximum flow will equal the size of the maximum matching.

#### 8. What is the basic idea of the Naïve String Matching algorithm?

**Ans:** The Naïve String Matching algorithm slides the pattern over the text one character at a time. At each position, it checks if the pattern matches the substring of the text. Though simple, it can be inefficient for large texts and repetitive patterns.

9. Define the Knuth-Morris-Pratt algorithm and mention one practical use-case in text editing tools. Ans: The Knuth-Morris-Pratt (KMP) algorithm is an efficient string matching method that avoids rechecking previously matched characters by using a prefix table (LPS array). It has a linear time complexity of O(n + m), making it faster than the naïve approach. A practical use-case is in "Find and Replace" features in text editors, where it quickly locates all occurrences of a word or phrase.

10.WhatistheprefixfunctionintheKMPalgorithm?Ans: The prefix function (or LPS array) tells us the longest proper prefix of the pattern that is also a suffix. Ithelps in deciding how many characters to skip in case of a mismatch. This pre-processing reduces the number ofcomparisons during the actual matching process.

11.WhatisthemainideabehindtheRabin-Karpalgorithm?Ans: The Rabin-Karp algorithm uses a hashing technique to compare the pattern with substrings of the text.Instead of checking character by character, it computes hash values and compares them. If a match is found inhash values, it then verifies the actual substring for confirmation.

12.WhyisRabin-Karpsuitablefordetectingmultiplepatterns?Ans: Rabin-Karp is efficient for multi-pattern matching because it can compute and compare hash values quickly.By precomputing hash values of all patterns, it can search for multiple patterns in a single pass through the text.This makes it ideal for applications like plagiarism detection.