d

Name: Jigar Siddhpura **SAPID:** 60004200155

DIV: C/C2 **Branch:** Computer Engineering

ADVANCE ALGORITHM (AA) RESEARCH PAPER BASED EXPERIMENT

ALGORITHM 1:

Algorithms/ Data structures:

Algorithm: Ford-Fulkerson Algorithm

Data Structure: Directed Graph

Theory: The Ford-Fulkerson algorithm is used to find the maximum flow in a network. It starts with an initial flow of zero and repeatedly augments the flow along the path from the source to the sink with available capacity, until no more augmenting paths exist.

Working:

- Start with an initial flow of zero.
- While there exists an augmenting path from source to sink:
- Find the augmenting path using depth-first search (DFS) or other graph traversal algorithms.
- Determine the minimum capacity along the augmenting path.
- Augment the flow along the path by the minimum capacity.
- Return the maximum flow when no more augmenting paths exist.

Applications:

- Network flow optimization: Maximizing the flow of goods through a transportation network.
- Traffic engineering: Optimizing traffic flow in road networks.
- Computer networks: Maximizing data flow in communication networks.
- Image segmentation: Used in algorithms like the max-flow min-cut theorem for image segmentation.

Complexity Analysis:

- The time complexity of the Ford-Fulkerson algorithm is O(E*f), where E is the number of edges in the graph and f is the maximum flow in the graph.
- The space complexity is O(V^2), where V is the number of vertices in the graph, due to the need to store the residual graph.

ALGORITHM 2:

Algorithms/ Data structures:

Algorithm: KD-Tree

Data Structure: Binary search tree (KD-tree)

Theory: Range Searching. The k-d tree, or k-dimensional binary search tree, was proposed by Bentleyin1975. It is a binary tree in which each record contains k keys, right and left pointers to its subtrees, and an index integer between 1 and k that indicates which key in the record is used for splitting.

Working: The KD-tree algorithm recursively partitions the space into smaller regions using hyperplanes perpendicular to the coordinate axes. Each node in the KD-tree corresponds to a region of the space, and the points are stored in the leaves. During a query, the algorithm traverses the KD-tree, visiting nodes whose regions intersect the query range and reporting points stored in those regions.

Applications:

- Geometric range searching - Nearest neighbour search involves finding the closest point(s) in a dataset to a given query point. In the context of KD-trees, nearest neighbour search is performed by traversing the KD-tree to locate the leaf node containing the query point and then backtracking to its parent nodes to search nearby regions. By keeping track of the closest point found so far and updating it when a closer point is discovered, the algorithm efficiently finds the nearest neighbour.
- Nearest neighbour search - Nearest neighbour search involves finding the closest point(s) in a dataset to a given query point. In the context of KD-trees, nearest neighbour search is performed by traversing the KD-tree to locate the leaf node containing the query point and then backtracking to its parent nodes to search nearby regions. By keeping track of the closest point found so far and updating it when a closer point is discovered, the algorithm efficiently finds the nearest neighbour.

Complexity Analysis: The construction time of a KD-tree is O(nlogn), and the query time for a rectangular range query is O(n^0.5+k), where n is the number of points and k is the number of reported points.

Reference Papers:

- Paper 1

Paper title: An Approach Based on Ford-Fulkerson Algorithm to Optimize Network Bandwidth Usage

Publication Name & Year: Brazilian Symposium on Computing Systems Engineering, 2015

- Paper 2

Paper title: Application of Ford-Fulkerson Algorithm to Maximum Flow in Water Distribution Pipeline Network

Publication Name & Year: International Journal of Scientific and Research Publications, Volume 8, Issue 12, 2018

- Paper 3

Paper title: Range Searching using KD tree

Publication Name & Year: International Journal of Scientific and Research Publications, Volume 6, Issue 5, 2005

- Paper 4

Paper title: Range Searching in Moderate Dimensions k-d Trees

Publication Name & Year: Proc. 33rd Sympos. Computer Geom, 2017.

- Paper 5

Paper title: Analysis of range search for random k-d trees

Publication Name & Year: Acta Informatica, 2001