Graph Algorithms Project Report

Group 40

February 15, 2025

Graph Algorithms Overview

1. Dijkstra's Algorithm

Dijkstra's algorithm is used to find the shortest paths from a source node to all other nodes in a weighted graph. It uses a priority queue to greedily select the closest vertex.
Time Complexity: $O(V^2)$ (using arrays) or $O(E + V \log V)$ (with a priority queue).

```
def dijkstra(graph, src):
    pass
```

2. Kruskal's Algorithm

Kruskal's algorithm is used to find the Minimum Spanning Tree (MST) of a graph by sorting edges and using the Union-Find data structure to avoid cycles.
Time Complexity: $O(E \log E)$.

```
def kruskal(edges, nodes):
    pass
```

3. Floyd-Warshall Algorithm

The Floyd-Warshall algorithm finds shortest paths between all pairs of nodes using dynamic programming.
Time Complexity: $O(V^3)$.

```
def floyd_warshall(graph):
    pass
```

Conclusion

This project demonstrates the implementation of key graph algorithms and their complexities while presenting them via a website.

Student Information

ID	Name	Roll Number
1	Simran Sesha Rao	2022A7PS0002H
2	Simran Singh	2022A7PS0003H
3	Shreya Kunjankumar Mehta	2022A7PS0115H
4	Sukhbodhanand Tripathi	2022A7PS0187H
5	Granth Bagadia	2022A7PS0217H