GPU Information

✅ GPU detected: NVIDIA GeForce RTX 4070

Memory: 12.88 GB

CUDA Version: 12.6

CS Educational Assistant

Welcome, Niveditha !

Logout

Upload computer science textbooks to build a knowledge base for guided learning.

Initialize/Restore Database

Upload PDF Textbook

Drag and drop file here

Limit 200MB per file • PDF

Browse files

Knowledge Base Stats

Total Q&A pairs: 1861

Backup to MongoDB

Load DeepSeek Model

📝 CraftNotes

Computer Science Educational Assistant

This assistant is designed to help computer science students learn concepts through guided discovery. Instead of giving direct answers, it provides learning roadmaps with resources and hints.

How to I Code Dikjistra's algorithm

To effectively implement Dijkstra's algorithm, here's a step-by-step guide:

1. Problem Understanding

- **Objective**: Find the shortest path from a starting vertex to all other vertices in a graph.
- **Constraints**: Graph may not be fully connected (no cycles), and edge weights may be positive, negative, or zero.
- **Edge Cases**: Multiple edges, empty edges (no outgoing edges), unreachable vertices.

2. Key Concepts

- **Adjacency List**: Represents each vertex with a list of tuples containing the connected vertex and edge weight.
- **Priority Queue**: Heaps implemented via heapq in Python to manage the selection of next vertices.
- **Distance Table**: Tracks the shortest distance from the start vertex to each vertex.
- **Previous Vertex Tracking**: Helps reconstruct the path once the end vertex is found.

3. Algorithm Design

1. **Initialization**

    - Read input graph and target vertex.
    - Initialize distance table with infinity for all vertices except start vertex.
    - Set start vertex distance to zero.
2. **Priority Queue Setup**
    - Push the start vertex into the priority queue.
    - Mark it as visited.
3. **Main Loop**
    - Extract vertex with smallest distance from priority queue.
    - If extracted vertex is target, break.
    - For each neighbor:
        - Calculate new distance.
        - Update if shorter.
        - Push neighbor into priority queue.
4. **Termination**
    - Return shortest path details.

4. Implementation Steps

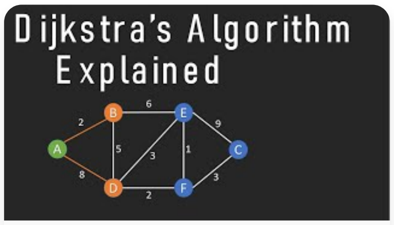
1. **Data Structures**
    - Use heapq for priority queue.
    - Adjacency list for graph representation.
2. **Reading Input**
    - Parse edges into adjacency list format.
    - Determine target vertex.
3. **Algorithm Execution**
    - Track distances and predecessors.
    -

Recommended Videos

Here are some educational videos that might help you learn more about this topic:

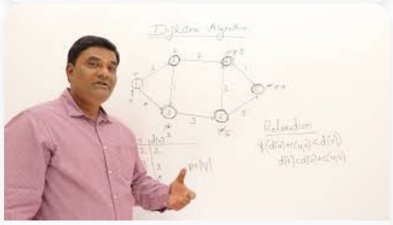
1. Dijkstras Shortest Path Algorithm Explained | With Exampl... by FelixTechTips
2. 3.6 Dijkstra Algorithm - Single Source Shortest Path - Gr... by Abdul Bari

3. L-4.10: Dijkstra's Algorithm - Single Source Shortest... by Gate Smashers
4. Lec-42 Dijkstra Algorithm In Hindi | Single Source Shorte... by Start Practicing



Dijkstras Shortest Path Algorithm Explained | With...

FelixTechTips



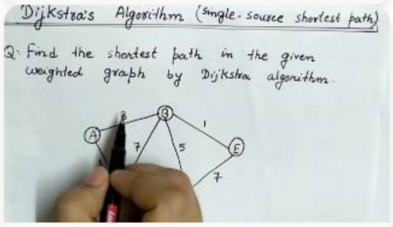
3.6 Dijkstra Algorithm - Single Source Shortest Path - Gr...

Abdul Bari

1.1M+ Views
Introduction to Dijkstra's Algorithm
Single Source Shortest Path

L-4.10: Dijkstra's Algorithm - Single Source Shortest...

Gate Smashers


Dijkstra's Algorithm (Single-source shortest path)
Q: Find the shortest path in the given weighted graph by Dijkstra algorithm.

Lec-42 Dijkstra Algorithm In Hindi | Single Source Shorte...

Start Practicing

Reset Chat

Ask a computer science question...