Dijkstra’s and Kruskal’s Algorithms

Ali Akbari

June 2025

Dijkstra’s Algorithm Example

- **Problem**: Shortest path from A to D: A→B (4), A→C (2), B→D (5), C→D (8).
- Update distances based on shortest known paths.
- **Step-by-Step**:
  1. Start: A:0, B:∞, C:∞, D:∞.
  2. Visit A: B:4, C:2.
  3. Visit C: D:10 (via C).
  4. Visit B: D:9 (via B).
  5. Path: A→B→D (cost 9).

Dijkstra's Algorithm

- Finds shortest path in weighted graph using priority queue.
- Always explores node with smallest known distance.
- **Time Complexity**: $O((V + E) \log V)$.
- **Space Complexity**: $O(V)$.

Kruskal’s Algorithm Example

- **Problem**: Minimum road length for cities: A-B (4), A-C (2), B-D (5), C-D (8), B-C (3).
- Select smallest edges, avoid cycles.
- **Step-by-Step**:
  1. Sort: A-C (2), B-C (3), A-B (4), A-B (4), C-D (8).
  2. Take A-C (2), B-C (3), B-D (5).
  3. Skip A-B (cycle).
  4. MST: A-C, B-C, B-D (total 10).

Kruskal’s Algorithm

- Builds minimum spanning tree by selecting smallest edges.
- Uses union-find to avoid cycles.
- **Time Complexity**: $O(E \log E)$ (sorting edges).
- **Space Complexity**: $O(V)$.