Dijkstra Algorithmus

Definition
- Kürzester Pfad
  - Single-Source Shortest Path
  - Von Startknoten zu allen Knoten

Voraussetzungen
- Positive Kantengewichte
- Keine negativen Gewichte
- Gerichteter/ungerichteter Graph

Datenstrukturen
- Distanz-Array (# initial)
- Vorgänger-Array
- Prioritätswarteschlange (Min-Heap)
  - Fibonacci-Heap optimal

Pseudocode
- initialize(dist, pred)
- Q = Prioritätswarteschlange
- while Q nicht leer
  - u = extract_min(Q)
  - for v in Nachbarn(u)
    - if dist[v] > dist[u] + w(u,v)
      - dist[v] = dist[u] + w(u,v)
      - pred[v] = u

Eigenschaften
- Greedy-Algorithmus
- Korrektheit durch Dreiecksinqualität
- Anwendungen
  - Routing (OSPF)
  - Netzwerkoptimierung
  - Kartenapps

Laufzeitkomplexität
- O(|V| log |V| + |E|) [Fibonacci-Heap]
- O(|V|^2) [Array-basiert]
- O(|E| log |V|) [Binär-Heap]

Algorithmus-Schritte
- Initialisierung
  - dist[start] = 0
  - dist[andere] = #
  - Vorgänger = null
- Schleife
  - Extrahiere Knoten u mit min. dist[u]
  - Für jeden Nachbarn v von u
    - Relax: dist[v] = min(dist[v], dist[u] + gewicht(u,v))
    - Update Vorgänger[v] = u
  - Bis alle Knoten besucht

© 2026 by Kleinhirn.eu