Dijkstra Implementation Using Java (Assignment - 1)

Abdur Rouf

September 2025

1 Pseudo code

From each Pair, the shortest path algorithm (Dijkstra) extracts the destination node and runs `calculateShortestPath(destination)`. The method computes the shortest path tree rooted at the destination and returns both distances (the minimum cost from every node to the destination) and paths (the predecessor of each node along its shortest route).

Function `calculateShortestPath(`*destination*`):`
- **forall** $v \in V \setminus \{destination\}$ **do**
  - $distances\{v\} \leftarrow \infty$;
- **end**
- $distances\{destination\} \leftarrow 0$;
- $paths\{destination\} \leftarrow destination$;
- $visited \leftarrow \emptyset$;
- $pq \leftarrow \{(destination, 0)\}$;
- **while** $pq \neq \emptyset$ **do**
  - $v \leftarrow$ node in pq with smallest $distances\{v\}$;
  - $visited \leftarrow visited \cup \{v\}$;
  - **forall** $u \in V$ **do**
    - $weight \leftarrow graph.getWeight(v, u)$;
    - **if** $weight > 0$ ***and*** $u \notin visited$ ***and*** $distances\{v\} + weight < distances\{u\}$ **then**
      - $distances\{u\} \leftarrow distances\{v\} + weight$;
      - $pq \leftarrow pq \cup \{(u, distances\{u\})\}$;
      - $paths\{u\} \leftarrow v$;
    - **end**
  - **end**
- **end**
- **return** $\{distances, paths\}$;