

Shortest Paths: Dijkstra's Algorithm

Problem: Given a graph (directed or undirected) with non-negative edge weights, compute shortest paths from given source node, s , to all nodes.

- “Shortest” = sum of weights along path is smallest.
- For each node, keep estimated distance from s , ...
- ...and of preceding node in shortest path from s .

```
PriorityQueue<Vertex> fringe;  
For each node v { v.dist() =  $\infty$ ; v.back() = null; }  
s.dist() = 0;  
fringe = priority queue ordered by smallest .dist();  
add all vertices to fringe;  
while (! fringe.isEmpty()) {  
    Vertex v = fringe.removeFirst ();  
  
    For each edge (v,w) {  
        if (v.dist() + weight(v,w) < w.dist())  
            { w.dist() = v.dist() + weight(v,w); w.back() = v; }  
    }  
}
```

Example

