CSc 345 — Analysis of Discrete Structures
(McCann)

Single Source Shortest Path (SSSP) Algorithms

Dijkstra's Algorithm

*Note: This algorithm is **not** guaranteed to work in the presence of negative edge weights.*

Legend:

source	the source (starting) vertex
$d(a, b)$	the path cost (distance) from vertex a to vertex b
$w(x, y)$	the weight of the edge connecting vertices x and y
Known	the set of vertices whose shortest paths from the source vertex are known
Fringe	the set of vertices that we know we can reach from the source

```
Set d(source,source) = 0 and d(source,x) = +infinity, for all other vertices x
Initialize Known to contain source
Initialize Fringe to contain the vertices adjacent to source
So long as Fringe is not empty:
    Find the fringe vertex f that has the smallest d(source,f)
             [ where d(source, f) = d(source,t) + w(t,f), where t is Known ]
    Move f from Fringe to Known
    Add unKnown, unFringe vertices that are adjacent to f to the Fringe
    Update Fringe data if necessary
```

Bellman-Ford Algorithm

Legend:

source	the source (starting) vertex
V	vertex set of the graph
$d(a, b)$	the path cost (distance) from vertex a to vertex b
$w(x, y)$	the weight of the edge connecting vertices x and y

```
Set d(source,source) = 0 and d(source,x) = +infinity, for all other vertices x
Loop (|V| - 1) times:
    For each edge (x,y) of the graph:
        if d(source,x) + w(x,y) < d(source,y) then
            Let d(source,y) <-- d(source,x) + w(x,y)
For each edge (x,y) of the graph:
    if d(source,x) + w(x,y) < d(source,y) then
        Indicate that algorithm failed due to a negative-weight cycle
Indicate that the algorithm completed successfully
```