The Bellman–Ford Algorithm

UNLV: Analysis of Algorithms Lawrence L. Larmore

The Single Source Minimum Path Problem

We are given a weighted directed graph $G = (V, E, W)$ with a designated source vertex $s$. That is, if $e = (u, v) \in E$, then $W(e) = W(u, v)$ is the weight of the edge $e$. A solution to the problem consists of arrays $\{V[v]\}_{v \in V}$ and $\{\text{back}[v]\}_{v \in V}$, such that $V[v]$ is the minimum weight of any path from $s$ to $v$, while $\text{back}[v]$ is the next-to-the-last vertex of one of those minimum paths. There is no solution if $G$ has a negative cycle.

\[
\text{for all } v \text{ in } V \\
\quad \text{back}[v] := \ast \\
\quad V[v] := \text{infinity} \\
\text{endfor} \\
V[s] := 0 \\
\text{altered} := \text{true} \\
\text{while (altered)} \\
\quad \text{altered} := \text{false} \\
\quad \text{for all } e = (u,v) \text{ in } E \\
\quad \quad \text{if } (V[u] + W(e) < V[v]) \\
\quad \quad \quad V[v] := V[u] + W(e) \\
\quad \quad \quad \text{back}[v] := u \\
\quad \quad \quad \text{altered} := \text{true} \\
\quad \text{endif} \\
\text{endfor} \\
\text{ endwhile}
\]

The running time of the Bellman-Ford algorithm is $O(nm)$. If $\ell$ is the length of the longest minimum weight path found, the above code runs in only $O(\ell m)$ time. If $G$ has a negative cycle, the above code will never halt.
The code below contains protection against this. If the while loop executes \( n \) times and some value of \( V \) is altered at the \( n^{th} \) iteration, there must be a negative cycle.

```plaintext
for all v in V
    back[v] := *
    V[v] := infinity
endfor
V[s] := 0
altered := true
numiterations := 0
while (altered and (numiterations < n))
    altered := false
    numiterations := numiterations + 1
    for all e = (u,v) in E
        if (V[u] + W(e) < V[v])
            V[v] := V[u] + W(e)
            back[v] := u
            altered := true
        endif
    endfor
endwhile
if (altered)
    Write('There is a negative cycle.')
endif
```