The Bellman–Ford Algorithm

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.

```plaintext
for all v in V
    back[v] := *
    V[v] := infinity
endfor

V[s] := 0
altered := true
while (altered)
    altered := false
    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
```

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.
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

Output of the Bellman Ford Algorithm

<table>
<thead>
<tr>
<th></th>
<th>s</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>0</td>
<td>7</td>
<td>1</td>
<td>7</td>
<td>9</td>
<td>∞</td>
</tr>
<tr>
<td>back</td>
<td>s</td>
<td>s</td>
<td>s</td>
<td>s</td>
<td>⊥</td>
<td>⊥</td>
</tr>
</tbody>
</table>

The first array shows the variables of the Bellman-Ford algorithm after one iteration, showing values of only paths of length 1 are considered, while the second array shows the values if only paths of length at most 2 are considered. After one execution of the main loop of the algorithm, the values of V will be no greater than those shown, but possibly smaller, depending on the order of the visitation of the edges.

The last array shows the variables of the Bellman-Ford algorithm if only paths of length at most 4 are considered. These will be the values after three iterations of the main loop of the algorithm, since no smallest path has length greater than 3.