

# AD [HA] zum 18. 12. 2013

Arne Struck, Lars Thoms

17. Dezember 2013

1.:

```
BellmanFord_modified(G,s):
    InitializeSingleSource(G,s)
    for i = 1, ..., |V| - 1
        nochanges = false
        for all edges(u, v) in E
            distTmp = v.dist
            Relax(u,v)
            if v.dist < distTmp
                nochanges = true
        if nochanges = true
            return true
```

Die Anpassung durch `nochanges` bewirkt eine Terminierung einen Durchlauf nachdem alle kürzesten Kantenpfade gefunden sind. Dies geschieht, da nach  $m$  Durchläufen alle kürzesten Kantenpfade entdeckt sind. Die Schleife wird noch einmal durchlaufen und hier wird festgestellt, dass insgesamt keine Änderungen an einem der Pfadgewichte vorgenommen wurde. Also wird der Algorithmus darauf Terminieren. Weitere Terminierungen sollten nicht sinnvoll sein, da sie durch den spezifizierten Input (kein negativen Zyklen) obsolet geworden sind. Der endgültige `return` kann auch weggelassen werden, da ein anderer `return` auf jeden Fall erreicht wird.

2.:

```
DAG-Shortest-Path(G,s):
    sort G.V topologically
    InitializeSingleSource(G,s)
    for each u in G.V % now in topological order
        for each v in Adj(u)
            Relax(u,v)
```

Die innere Schleife wird  $|E|$  mal durchlaufen, da jede Kante adressiert wird.

Die äußere Schleife wird  $|V|$  mal durchlaufen (jeder Knoten ein mal).

Durch die vorherige topologische Sortierung ist garantiert, dass der Algorithmus nicht mehrmals durchlaufen werden muss (da die Knoten jetzt in einer linearen Richtung vorliegen).

Damit ist gezeigt, dass die Laufzeit  $\mathcal{O}(|E| + |V|)$  beträgt.

**3.:**

**TODO**

**4.:** a)

**TODO**

b)

**TODO**

**5.:** a)

**TODO**

b)

**TODO**

**6.:**

**TODO**