Sei $G = (V, E)$ ein DAG und $s \in V$. Sei **vor** die Relation in der topologischen Sortierung.

DAG-Bellman-Ford

```
1: procedure DAG-Bellman-Ford(G, s)
2:     InitializeSingleSource(G, s)
3:     list =Topological-Sort(G)
4:     start = 0
5:     while list[start] ≠ s do
6:         start ← start + 1
7:     end while
8:     for v in list[start + 1 : |V|] do
9:         for all (u, v) in G.E do
10:            Relax(u, v)
11:        end for
12:    end for
13: end procedure
```

Behauptung. *In Algorithmus wird* Relax *höchstens $|E|$-mal aufgerufen.*

Beweis. Nach Zeile 3 befinden sich alle Knoten von G in topologischer Reihenfolge in *list*. Die innere Schleife wird für jeden Knoten v mit s **vor** v einmal aufgerufen. Dies sind höchstens $|V| - 1$ Knoten. In dieser Schleife wird Relax einmal für jede Kante, welche zu diesem Knoten führt, aufgerufen. Da jede Kante zu genau einem Knoten führt, wird Relax für jede Kante höchstens einmal aufgerufen. Dies sind höchstens $|E|$ Ausführungen von Relax. □

Behauptung. *Algorithmus löst das Single-Source-Shortest-Path-Problem.*

Beweis. Sei $t \in V$.

- $t = s$
  In InitializeSingleSource wird die korrekte Distanz (0) gesetzt.

- t **vor** s
  Da G ein DAG ist und es einen Pfad von t nach s gibt, existiert kein Pfad von s zu t (sonst wäre G nicht azylisch). t steht vor s in *list*. Da nur der Teil von *list* ab s bearbeitet wird, wird die Distanz von s nach t nicht neu gesetzt, nachdem sie mit ∞ initialisiert wurde.

- $t \neq s \wedge \neg(t$ **vor** $s) \wedge \neg(s$ **vor** $t)$
  Es gibt keinen Pfad zwischen s und t. Falls t vor s in *list* steht, wird die Schleife nicht auf t ausgeführt, sonst wird wie in dem nächsten Punkt vorgegangen.

- s **vor** t

  1. Induktionsanfang: $(s, t) \in E$
     Sobald die äußere Schleife bei t angekommen ist, wird Relax für diese Kante

aufgerufen und setzt die korrekte Distanz von s zu t (gegeben durch $w(s,t)$), da $t.dist = \infty > s.dists + w(s,t) = w(s,t)$ galt.

2. Induktionsannahme: Bei Erreichen eines Knotens k in $list$ wird die korrekte Distanz zu s gesetzt.

3. Induktionsschritt $(s,t) \notin E$
   Dann gibt es eine Menge von Knoten

$$K = \{k \in V \mid s \textbf{ vor } k \wedge (k,t) \in E\} \tag{1}$$

   Da für alle $k \in K$ gilt, $(k,t) \in E$, gilt auch k **vor** t. Bevor die äußere Schleife t erreicht, hat sie bereits alle k bearbeitet. Nach der Induktionsannahme haben alle k bereits die korrekte Distanz zu s. Während t in der Schleife gewählt ist, werden die Kanten von $k \in K$ durchlaufen und $t.dist = \min(k.dist + w(k,t))$ gesetzt. Ist $K = \emptyset$, so bleibt $t.dist = \infty$.

□