

# Alg. Kürzeste-Pfade im DAG

1. Sortiere  $V$  topologisch

2. Initialisiere  $(G, s)$

3. Für jeden Knoten  $u \in V$  in der sortierten Reihenfolge

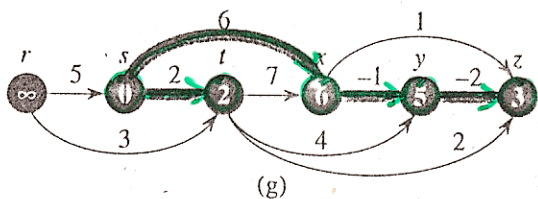
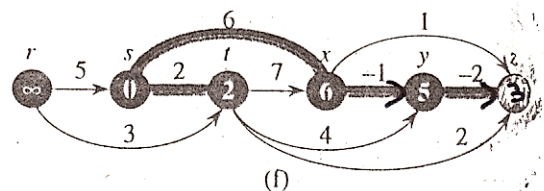
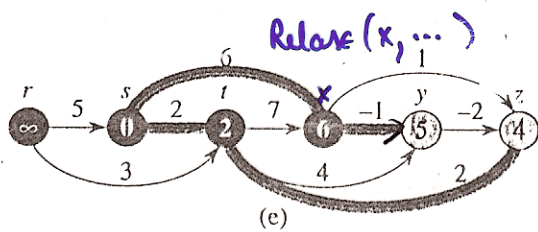
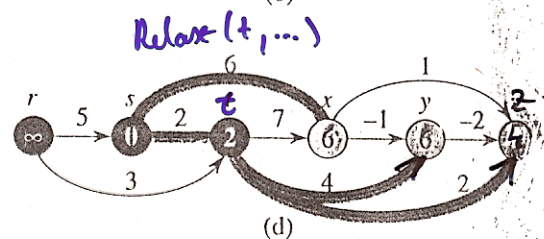
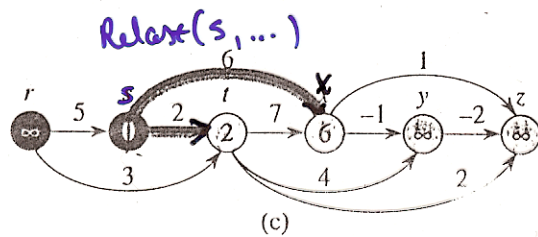
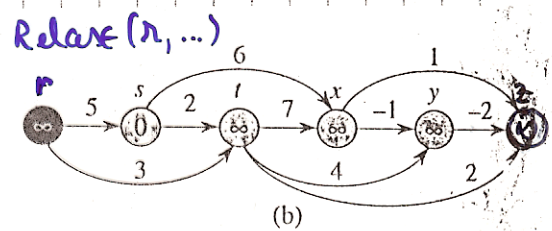
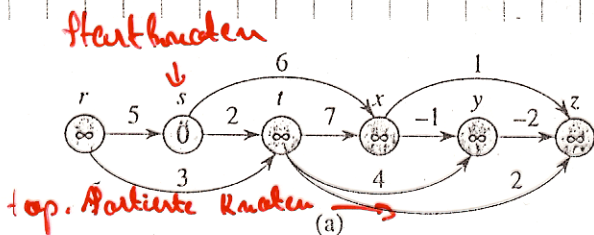
Rufe RELAX  $(u, v, c(u, v))$  für jedes  $v \in \text{Adj}^+(u)$  auf

mit RELAX  $(u, v, c(u, v))$ :

if  $d(v) > d(u) + c(u, v)$

then  $d(v) \leftarrow d(u) + c(u, v)$

$F(v) \leftarrow u$



Kürzeste - Pfad - Baum

Algorithmus zur Berechnung der kürzesten Pfade in einem gerichteten azyklischen Graphen. Die Knoten sind von links nach rechts topologisch sortiert. Der Startknoten ist  $s$ . Die Werte von  $d$  sind in jedem Knoten eingetragen. Schattierte Kanten zeigen die  $F$ -Werte. (a) Der Zustand vor der ersten Iteration der for-Schleife in der Zeile 3. (b)-(g) Der Zustand jeweils nach einer weiteren Iteration der for-Schleife. Der neu geschwätzte Knoten wurde in der jeweiligen Iteration für  $u$  benutzt. Die in Teil (g) gezeigten Werte sind die endgültigen.