Topologische Sortierung

- Das Problem des topologischen Sortierens auf einem gerichteten, azyklische Graphen besteht darin, alle Knoten so in eine Reihenfolge zu bringen, dass für je zwei Knoten u und v gilt: Wenn es einen Weg von u nach v gibt, dann kommt u vor v in der Knotenfolge der Wegnotierung.
- Alle Graphen, die keine Zyklen enthalten (azyklische Graphen), sind topologisch sortierbar. Wenn zyklische Abhängigkeiten bestehen, existiert keine topologische Sortierung.
- Einfacher Linearisierungsalgorithmus: Solange noch Knoten vorhanden sind, wähle einen vorgängerlosen Knoten aus, trage diesen in eine Ausgabeliste ein und entferne den Knoten aus der Menge.

Tiefen- und Breitensuche

- **Ziel:** Finden aller von einem Startknoten aus erreichbaren Knoten
- Zwei prinzipielle Strategien der Graphensuche:
  - **Tiefensuche (DFS - Depth-First Search):** Wenn es von einem gerade besuchten Knoten u eine ausgehende Kante gibt, dann wird zunächst der Zielknoten dieser Kante untersucht (und diese Strategie dann auf diesen Knoten angewendet) bevor die Nachbarknoten von u untersucht werden.
  - **Breitensuche (BFS - Breadth-First Search):** Ausgehend von einem gerade besuchten Knoten u werden erst alle direkten Nachfolgeknoten von u untersucht, um dann deren Nachfolgeknoten zu untersuchen, usw.
- Beide Strategien sind nichtdeterministisch, d.h. für das Abarbeiten der Nachfolger eines Knotens ist keine feste Reihenfolge vorgegeben und je nachdem, welche Reihenfolge in einem konkreten Suchvorgang gewählt wurde, können sich für den gleichen Eingabegraphen verschiedene Spannbäume ergeben.

Dijkstra-Algorithmus

- Protokollierung der aktuellen Randknotenmenge und den zugehörigen Auswahlknoten:
  1. Startknoten in der Form (Knoten, 0, -) als Auswahlknoten notieren.
  2. Direkte Nachfolger des aktuellen Auswahlknotens in der Form (Ziel, Entfernung, Start) in Randknotenmenge aufnehmen.
     - Wurde der Zielknoten bereits als Auswahlknoten gewählt, wird er nicht erneut in Randknotenmenge aufgenommen.
     - Zur Berechnung der Entfernung wird der Kantenwert zu der Entfernung des aktuellen Knotens addiert.
     - Ist der Zielknoten bereits in der Randknotenmenge enthalten, wird das neue Tupel nur aufgenommen, wenn es eine kürzere Entfernung besitzt.
  3. Wenn die Randknotenmenge leer ist, endet der Algorithmus.
  4. Ein Tupel mit kürzester Entfernung markieren.
  5. Markiertes Tupel als neuen Auswahlknoten notieren und mit Punkt 2 fortfahren.
- Aus diesem Ablaufprotokoll lassen sich die kürzesten Wege vom Startknoten zu jedem beliebigen Knoten und die jeweils zu durchlaufende Knotenfolge ablesen.

Übung 4

gewählt	**Randknotenmenge**
$(1, 0, -)$	$\{(2, 7, 1), (6, 4, 1), (7, 1, 1)\}$
$(7, 1, 1)$	$\{(2, 7, 1), (4, 7, 7), (5, 3, 7), (6, 4, 1)\}$
$(5, 3, 7)$	$\{(2, 7, 1), (4, 5, 5), (6, 4, 1)\}$
$(6, 4, 1)$	$\{(2, 7, 1), (4, 5, 5)\}$
$(4, 5, 5)$	$\{(2, 7, 1), (3, 6, 4)\}$
$(3, 6, 4)$	$\{(2, 7, 1)\}$
$(2, 7, 1)$	$\emptyset$

Zielknoten	**Pfadlänge**	**kürzester Pfad**
1	0	[1]
7	1	[1, 7]
5	3	[1, 7, 5]
6	4	[1, 6]
4	5	[1, 7, 5, 4]
3	6	[1, 7, 5, 4, 3]
2	7	[1, 2]