Lösungen zum Thema Graphen

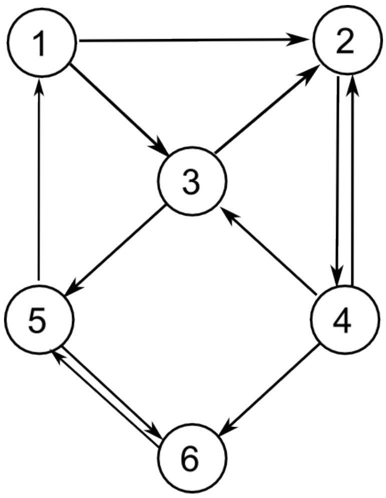
(1) Knoten $1, 2, 3, 4$ und folgende Kanten mit Gewichten: $c(\{1, 2\}) = 1$, $c(\{2, 3\}) = 1$, $c(\{3, 4\}) = 1$ und $c(1, 4) = 2$. Bilde den kürzeste-Wege-Baum von 1 aus. Benutzte Kanten: $\{1, 2\}, \{2, 3\}, \{1, 4\}$ mit Gewicht 4. Der minimale Spannbaum besteht aus den drei Kanten mit Gewicht 1 (Gesamtgewicht 3).

(2) (a) Knoten $1, 2, 3$ und folgende Kanten mit Gewichten: $c(\{1, 2\}) = 3$, $c(\{2, 3\}) = -2$, und $c(\{1, 3\}) = 2$. Startknoten sei Knoten 1.

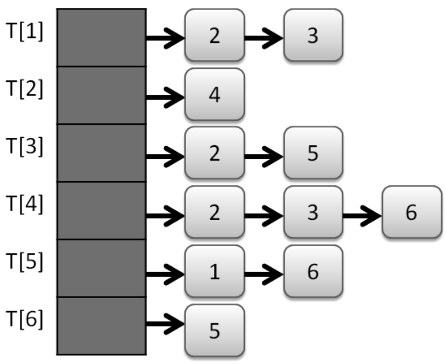
(b) Knoten $1, 2, 3$ und folgende Kanten mit Gewichten: $c(\{1, 2\}) = 1$, $c(\{2, 3\}) = -1$, und $c(\{1, 3\}) = 9$. Startknoten sei wieder Knoten 1.

(c) bei ungerichteten Graphen gilt: negative Kante = negativer Kreis (Kante beliebig oft vor und zurück).

(3) (a)



(b)



(c) Die dritte vorgestellte Möglichkeit ist die Adjazenzmatrixdarstellung. In einer $|V| \times |V|$-Matrix A wird an der Stelle $A_{i,j}$ eine 1 gesetzt, wenn zwischen den Knoten i und j eine Kante existiert, andernfalls eine 0.
Vorteile:

- Problem "Existiert Kante von i nach j?" in konstanter Zeit lösbar
- Einfaches Hinzufügen und Löschen von Kanten möglich
- manche Graphenoperationen direkt durch Matrixoperationen durchführbar (z.B. "Existiert ein Pfad von A nach B?" durch Potenzieren der Adjazenzmatrix)

Nachteile:

- verhältnismäßig hoher Speicherverbrauch bei dünnen Graphen