

Wissensfragen: Lösungen

(1) mögliche Lösungen:

- Kruskal: es wird immer die Kante mit dem niedrigsten Gewicht ausgewählt, die Teil eines minimalen Spannbaums sein kann.
- Prim: es wird immer die leichteste Kante, die den bisherigen MST mit einem neuen Knoten verbindet, ausgewählt.
- Dijkstra: Durch die Verwendung einer Prioritätswarteschlange werden immer die ausgehenden Kanten des Knotens mit der geringsten Distanz relaxiert.

(2) mögliche Lösungen

- (+) besseres Verhalten im Worst Case
- (+) stabil
- (-) im Durchschnitt eher langsamer als Quicksort
- (-) zusätzlicher Speicher benötigt (nicht in-place)

(3) Stabilität bedeutet, dass sich die Reihenfolge von gleichen Elementen beim Sortieren nicht ändern. Anwenden lässt sich diese z.B. bei Tupeln, die lexikographisch geordnet werden (d.h. zuerst nach dem ersten Wert, bei Gleichheit nach dem zweiten). Ein Beispiel hierfür wären Tupel der Form (Monat, Tag) für einen Tag innerhalb eines Jahres. Man sortiert dabei zuerst nach dem zweiten Wert und danach mit einem stabilen Sortieralgorithmus nach dem ersten Wert. Bei Tupeln mit gleicher erster Komponente wird dann nämlich die ursprüngliche Reihenfolge beibehalten, welche durch Sortieren nach der zweiten Komponente erreicht wurde.

(4) Falsch, da binäre Suche nicht auf jeder Datenstruktur anwendbar ist: Gegenbeispiel: Verkettete Listen

(5) Falsch! Gegenbeispiel für den Leser.

(6) Falsch! Gegenbeispiel, ein Graph bei dem die letzte Kante zum Zielknoten größer ist als die Summe aller Kantengewichte ohne diese Kante. Der Weg zu dieser Kante führt nur über die Kante mit dem eindeutig minimalstem Gewicht.

Dijkstra2 wird diese Kante nie fortführen und deswegen nicht den längsten Pfad finden.