Fortgeschrittene Algorithmen und Datenstrukturen

Seminar 5

1. Dijkstra

Implementieren Sie den Algorithmus von Dijstra zur Lösung des Single-Source Shortest Path Problems. Erweitern Sie hierfür die Ausarbeitung aus Seminar 4. Achten Sie besonders darauf, dass die Struktur aus dem Pseudocode in Ihrer Implementierung klar ersichtlich ist.

Geben Sie als Ergebnis nicht nur die Längen der kürzesten Pfade, sondern den gesamten Dijkstra-Spannbaum graphisch aus.

2. Prim

Implementieren Sie den Algorithmus von Prim durch eine Anpassung der Relaxation. Entwickeln Sie dafür ein objektorientiertes Konzept zum einfachen Ersetzen der Relaxation.

Dieses Beispiel ist als Erweiterung des vorigen Beispiels gedacht.

In Bezug auf Algorithmen wie Dijkstra und Prim bezieht sich "Relaxation" auf den Prozess der Aktualisierung von Entfernungen oder Gewichten während der Ausführung des Algorithmus. Es ist eine zentrale Operation, die zur Ermittlung des kürzesten Pfades bzw. des minimalen Spannbaums beiträgt.

Relaxation im Dijkstra-Algorithmus:
Im Dijkstra-Algorithmus wird die Relaxation verwendet, um die bisher bekannten kürzesten Entfernungen zu den Knoten zu aktualisieren, wenn ein kürzerer Weg gefunden wird. Dies geschieht, indem die Distanz zum aktuellen Knoten plus das Gewicht der Kante zu seinem Nachbarn mit der aktuellen Entfernung zum Nachbarn verglichen wird. Wenn der neu berechnete Weg kürzer ist, wird die Entfernung aktualisiert und der Vorgängerknoten wird festgelegt. Dieser Vorgang wird wiederholt, bis alle kürzesten Pfade gefunden sind.

Relaxation im Prim-Algorithmus:
Im Prim-Algorithmus wird die Relaxation verwendet, um die Gewichte der Kanten im minimalen Spannbaum zu aktualisieren, wenn eine Kante mit einem kleineren Gewicht gefunden wird, um einen Knoten mit dem aktuellen minimalen Spannbaum zu verbinden. Anders als beim Dijkstra-Algorithmus geht es hier nicht um die Entfernungen zu den Knoten, sondern um die Gewichte der Kanten im minimalen Spannbaum.