

Abgabe zu Aufgabenblatt 9

Graph und Dykstra

Karanpal Singh Robert Rösler

6. Juni 2017

Zusammenfassung

In dieser Aufgabe wurde ein ungerichteter und gewichteter Graph einmal mit Adjazenzlisten und einmal mit Adjazenzmatrizen implementiert. Außerdem wurde der Dykstra Algorithmus zur Findung kürzester Pfade im Graphen implementiert.

Inhaltsverzeichnis

1	Graph-Interface	2
2	Entkopplung	3
3	Komplexitätsuntersuchung	4

1 Graph-Interface

Das Interface für die implementierten Graphen:

1. `void insert(E elem);`
Fügt einen neuen Knoten in die Knotenliste des Graphen ein.
2. `E remove(E elem);`
Entfernt Knoten mit Inhalt `elem` und löscht alle Kanten, die von diesem Knoten ausgehen.
3. `void addLink(E von, E nach, int costs);`
Fügt eine Kante von Knoten `"von"` nach Knoten `"nach"` mit den Kosten `"cost"` ein.
4. `List<Knoten<E>> getNeighbours(E elem);`
Liefert eine Liste aller Knoten, die mit dem Knoten `elem` Verbunden sind.
5. `public int getCost(E von, E nach);`
Liefert die Kosten der Kante von `"von"` nach `"nach"`.
6. `List<Knoten<E>> getKnoten();`
Liefert eine Liste aller Knoten des Graphen zurück, an dem die Methode aufgerufen wird.

2 Entkopplung

Folgende UML-Grafik zeigt die Struktur der Klassen:

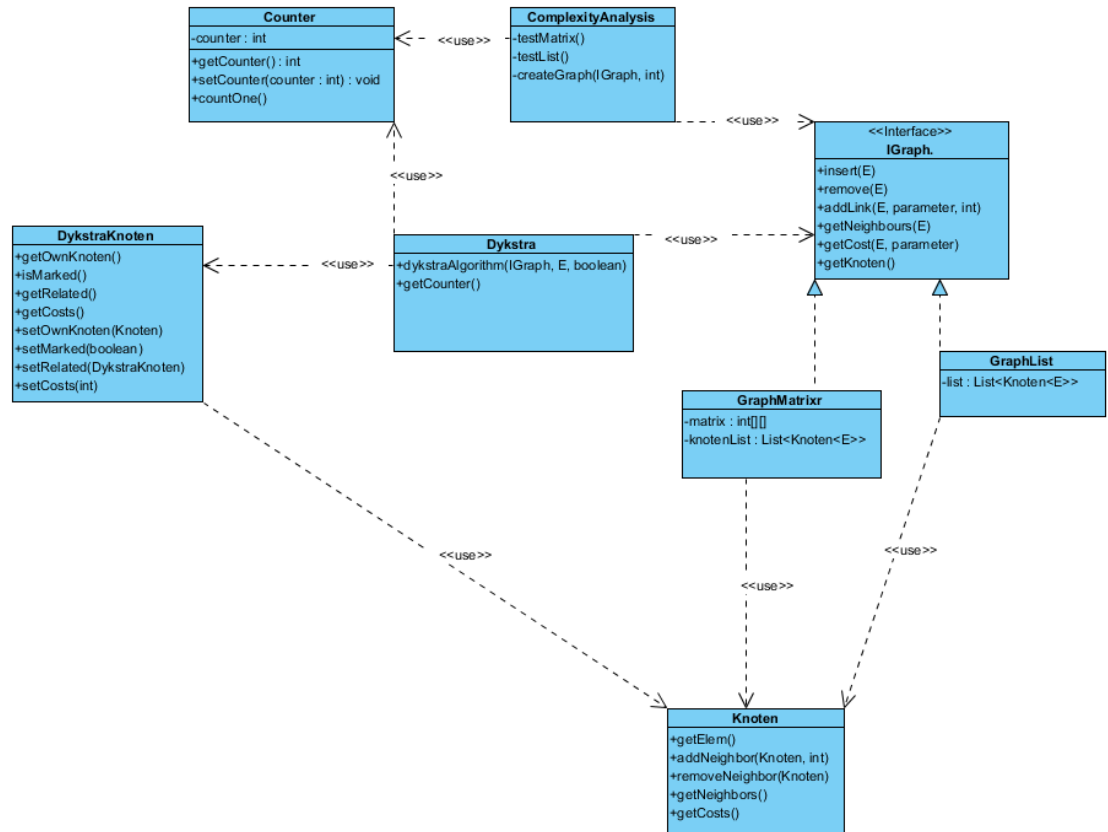


Abbildung 1: Programmstruktur

Die Graphen sind über ein gemeinsames Interface implementiert, lediglich die darunterliegende Verwaltungsstruktur für den Graphen unterscheidet sich. Über einen Wrapper für den Knoten, den DykstraKnoten werden die Extra-Informationen, die für die Berechnung des Dykstra-Algorithmus notwendig sind, verwaltet. Somit ist das Dykstra-Verfahren vom Graph entkoppelt und der Graph auch ganz allein für sich verwendbar. Der Zähler kann im Dykstra Algorithmus über einen Parameter zugeschaltet werden, kann also bei Bedarf auch nicht aktiviert werden, damit die Rechenleistung nicht beeinflusst wird.

3 Komplexitätsuntersuchung

Für die Komplexitätsuntersuchung werden Zähler in allen Schleifen platziert, in folgender Grafik werden die Werte für die Knotenanzahl $N = 10$, $N = 100$ und $N = 1000$, jeweils für die Listenimplementation und die Matriximplementation dargestellt.

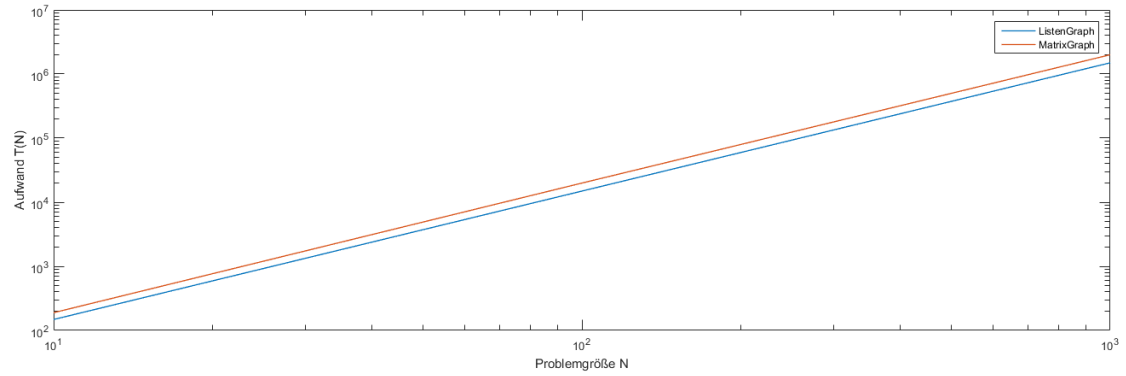


Abbildung 2: Aufwandsbestimmung