**Praktikum Graphentheoretische Konzepte und Algorithmen - Prof. Dr. Julia Padberg**

**Wintersemester 2018**

**Bearbeiter:** Tri Pham, Nhat Khanh Huy Tran; **Gruppe:** 5; **Team:** H

**Aufgabe 1: Visualisierung, Speicherung und Traversierung von Graphen**

**Aufgabenaufteilung:**

* Tri Pham:
  + Implementierung der Algorithmen und zum Teil Visualisierung des Graph.
  + Dateien: AlgoBFS.java, defaultStyleSheet
* N.K.Huy Tran:
  + Einlesen und Speichern von Graphen und zum Teil Visualisierung des Graph.
  + Dateien: GkaGraph.java, GkaUtils.java, defaultStyleSheet

**Quellenangaben:**

* GraphStream: <http://graphstream-project.org/>
* Die Implementierung des Algorithmus BFS stammt aus: <https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/>

**Bearbeitungszeitraum:** ungefähr 2 Wochen

**Aktueller Stand:**

**Beschreibung der Algorithmen:**

* Algorithmus zur Traversierung eines Graphen mittels BFS:

Gegeben sei ein Graph G mit einem Startknoten s. Der Algorithmus gibt auf die Konsole die Reihenfolge der Knoten laut BFS zurück.

* 1. Man markiere den Startknoten und füge in eine Liste hinzu.
  2. Man entferne den Knoten n, der an der ersten Stelle der Liste liegt und gebe n auf die Konsole zurück.
  3. Man ermittle alle Knoten in G, die zu n benachbart sind. Falls der Knoten noch nicht markiert ist, markiere ihn und füge ihn in der Liste hinzu.
  4. Falls die Liste leer ist, beendet der Algorithmus, sonst gehe zu Schritt 2.
* BFS-Algorithmus zur Ermittlung der Länge des kürzesten Weges zwischen zwei Knoten eines Graphen:

Gegeben sei ein Graph G mit zwei ausgezeichneten Knoten s und t. Der Algorithmus gibt die mindesten Anzahl der Kanten zurück, die benötigt sind, um s und t zu verbinden.

1. Man markiere s mit dist(s) = 0 (Abstand s-s) und füge s in eine Liste hinzu.
2. Man entferne den Knoten n, der an der ersten Stelle der Liste liegt. Man ermittle alle zu n benachbarte Knoten in G. Falls der Knoten noch keinen positiven Abstand hat, setze seiner Abstand dist(n) + 1.
3. Wenn t markiert wurde, gebe dist(n) + 1 als Ergebnis zurück, der Algorithmus wird beendet, sonst gehe zu Schritt 2. Wenn die Schleife beendet, bevor t markiert wurde, ist t nicht mit s über einen Weg verbunden.

* BFS-Algorithmus zur Ermittelung des kürzesten Wegs zwischen zwei Knoten eines Graphen:

Gegeben sei ein Graph G mit zwei ausgezeichneten Knoten s und t. Der Algorithmus gibt einen kürzesten Weg zurück, der s und t verbindet.

1. Man markiere t mit seinem mindesten Abstand von s dist(t) und speichere t in eine Liste.
2. Man ermittle einen Knoten, der zu dem zuletzt gespeicherten Knoten n benachbart sind und einen Abstand dist(n) – 1 von s hat. Man speichere den Knoten in der Liste.
3. Wenn der aktuelle Abstand 1 ist, folgt Schritt 4, sonst gehe zu Schritt 2.
4. Speichere s in der Liste. Der Algorithmus wird beendet. Die Reihenfolge der Knoten in der Liste entspricht den Weg von t nach s.

**Beschreibung der Datenstrukturen:**

**Wesentliche Entwurfsentscheidungen der Implementierung:**

Bei aller drei Algorithmen benutzen wir eine List bzw. ein queue, um die Knoten zu durchiterieren. Mithilfe der Liste kann man während der Durchläufe die neuen Knoten einfach hinzufügen und damit sich schnell die Reihenfolge der Untersuchung der Knoten entscheiden. Außerdem wenn ein Knoten untersucht wird, kann man der Knoten von der Liste entfernen, sodass man nicht derselbe Knoten noch mal untersuchen wird. Wenn die Liste leer wird, das heißt es gibt keine Knoten mehr zu untersuchen, ist der Algorithmus beendet.

**Umsetzung der Aspekte der Implementierung:**

**Dokumentation der Testfälle:**

**Beantwortung der Fragen:**

1. Was passiert, wenn Knotennamen mehrfach auftreten?

Vb

1. Wie unterscheidet sich der BFS für gerichtete und ungerichtete Graphen?

Vcc

1. Wie können Sie testen, dass Ihre Implementierung auch für sehr größe Graphen funktioniert?

vcc