BAI3-GKA WiSe23
Graphentheoretische Konzepte und Algorithmen

Praktikumsaufgabe 3 - Hierholzer

Dokumentation

GKA-Gruppe	
Team	
	Iryna Trygub
	Ansgar Deuschel
	Kristoffer Schaaf

Bearbeitete Themen in Stichpunkten:

Iryna Trygub	Graph Generator
Ansgar Deuschel	Hierholzer, Energiemessungen, Dokumentation
Kristoffer Schaaf	Tests, CVSWriter, Dokumentation

Inhaltsverzeichnis

1 Einleitung

2 Laufzeit

3 Dokumentation der Implementierung

3.1 Testen der ungeraden Knotengerade

In einem Graphen mit mindestens einem Knoten, welcher einen ungeraden Knotengrad hat, kann der Hierholzer Algorithmus keinen Erfolg haben. Ein Beispiel ist hier2 in den Tests zu finden.

Um sicherzustellen, dass dem Algorithmus ein valider Graph übergeben wird, wurde folgende Methode implementiert:

```
private static boolean graphIsValid(Graph graph) {
    ....
    List<Node> nodesWithFalseDegree = graph.nodes()
                            .filter(node -> node.getDegree() % 2 != 0)
                            .collect(Collectors.toList());
    for(Node node: nodesWithFalseDegree) {
        if(node.getEdgeBetween(node.getId()) == null) {
            return false;
        }
    }
    ...
}
```

Um zu prüfen ob Knoten einen ungeraden Kantengrad haben, werden diese Knoten zuerst alle in einer Liste gespeichert. Das ist notwendig, denn durch die genutzte Graphstream Bibliothek ensteht folgendes Problem: Angenommen es existiert ein Graph mit einem Knoten A und zwei Nachbarknoten B und C. Der Knoten A hat eine Schleife. Der Graph ist vollständig - A,B und C sind also alle miteinander verbunden (Abb. 7).

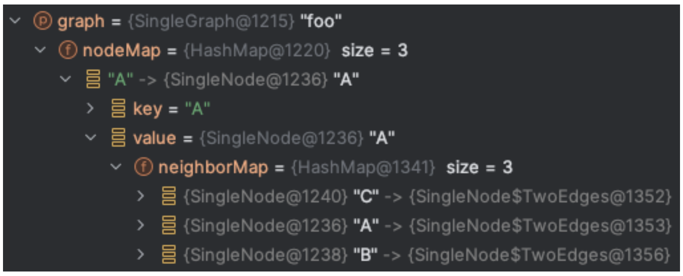

Abbildung 1: Ausschnitt des Graphen mit Loop aus dem Debugger

Wie nun im Debugger zu sehen1 hat der Knoten A nun nur drei Nachbarn. Zusätzlich hat er auch nur drei adjazente Kanten, AA, AB und AC. Das ist soweit alles korrekt. Der daraus resultierende Knotengrad ist allerdings ungerade und würde diesen Graph als invalide auswerten. Bei Betrachtung des Graphen hat dieser aber einen geraden Knotengrad.
Um dieses Problem zu umgehen wird in allen Kanten mit ungeradem Knotengrad also zusätzlich geprüft ob diese Kanten keine Schleifen sind. Erst wenn das der Fall ist, wird der Graph als invalide ausgewertet.

4 Tests

4.1 Fest definierte Testgraphen

4.1.1 False

Getestet wird zum einen mit verschiedenen fest definierten Testgraphen. Diese entsprechen bestimmten Vorgaben und sollten alle relevanten Edge Cases abdecken.

Für den Hierholzer Algorithmus dürfen keine gerichteten Graphen übergeben werden. Der erste Test prüft dies. Wird ein gerichteter Graph übergeben, beendet sich der Algorithmus und wirft eine Exception.

Der zweite zu testende Graph2 ist das Haus vom Nikolaus. Für diesen Graphen ist ein Eulerweg zu finden, allerdings kein Eulerkreis. Der Hierholzer Algorithmus schlägt hier also fehl. Beim ersten Durchlauf findet dieser den Kreis mit den Knoten A, B und C. Danach B, E, C, D. Anschließend können keine weiteren Kanten markiert werden ohne bisher markierte nochmal zu besuchen.
Der implementierte Algorithmus wirft in diesem Fall eine Exception.

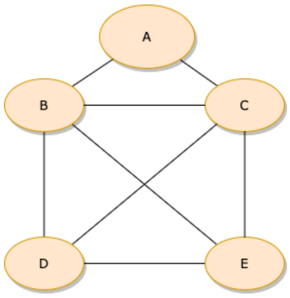

Abbildung 2: Haus vom Nikolaus - ungerade Knotengerade

Der dritte Graph3 besteht aus zwei Komponenten. Diese hängen nicht zusammen. Folglich muss der Algorithmus auch hier abbrechen, da dies nicht erlaubt ist. Wenn nicht alle Komponenten zusammenhängen, können auch nicht alle Kanten vom einem Startpunkt aus markiert werden.

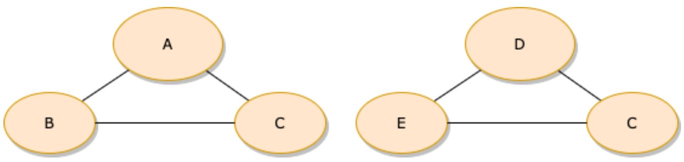

Abbildung 3: Ein Graph mit zwei nicht zusammenhängenden Komponenten

Es folgt ein Graph4 welcher mehr Knoten als Kanten besitzt. Auch in diesem Fall kann kein Eulerkreis zu finden sein. Wie im Beispiel zu sehen ist können hier der Start- und Endknoten niemals verbunden sein.

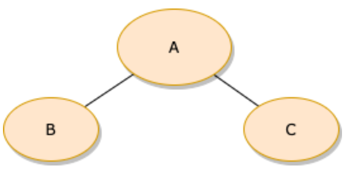

Abbildung 4: Ein Graph mit mehr Knoten als Kanten

4.1.2 True

Der erste Test in welchem der Hierholzer durchlaufen soll testet einen leeren Graphen. Wir geben vor, dass Graphen, welche weder Knoten noch Kanten haben, ein leeres Ergebnis liefern und keine Exception werden.

Der nächste Graph5 besteht aus einem einzigen Knoten. Der Algorithmus findet keine Kante, aber einen Startknoten. Da der Startknoten hierdurch auch zum Endknoten wird, wird auch hier keine Exception geworfen und der Algorithmus läuft durch.

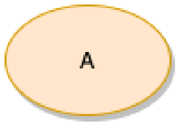

Abbildung 5: Ein Graph mit einem einzigen Knoten

Dieser Graph6 erinnert an eine Sanduhr. Zusätzlich hat er hat mehr Kanten als Knoten. Im Vergleich zum Test bei welchem mehr Knoten als Kanten im Graphen waren, können in diesem Graphen zwei Eulerkreise gefunden werden. Vorausgesetzt als Startknoten werden Knoten A und C genutzt enthält der erste Eulerkreis die Knoten A, B und C - der zweite C, D und E.

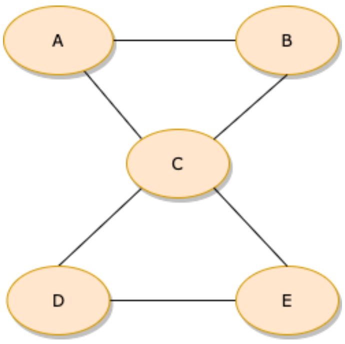

Abbildung 6: Ein Graph mehr Kanten als Knoten

Folgender Graph7 enthält eine Schleife. Auch in diesem Beispiel findet der Hierholzer Algorithmus zwei Eulerkreise. Der Knoten A ist durch eine Schleife mit sich selbst verbunden und bildet somit den ersten Eulerkreis. Der zweite Eulerkreis enthält die Knoten A, B und C.

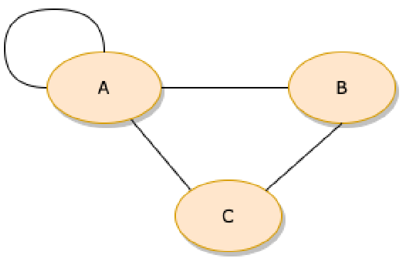

Abbildung 7: Ein Graph mit einer Schleife

Der letzte statische Graph8 ist ein Multigraph. Dieser enthält zwei Knoten, welche durch zwei Multikanten miteinander verbunden sind. Das hier ein Eulerkreis gefunden wird ist nach den letzten Tests offensichtlich. Dieser Test prüft hauptsächlich ob die Multigraphen der Bibliothek Graphstream richtig verarbeitet werden.

Abbildung 8: Ein Graph mehr Kanten als Knoten

Literatur