

Algorithmen und Programmierung

Übungsblatt 7

WS 2022/23

Dr. Felix Jonathan Boes

Benedikt Bastin, Ellen Bundschuh, Anna Höpfner, Gina Muuss, Adrian Oeyen, Felix Roth,
Thore Wolf

Ausgabe: 21.11.2022

Abgabe: keine Abgabe; Präsentation in der Übung

Aufgabe 1 (Breitensuche). In Ihrem Abgabe-Repository hat Flavius einen Branch mit dem Namen `Zettel_07` angelegt. Erweitern Sie die dort, unter Verwendung der Projektstruktur, Ihre Implementierung der Klasse `SimplerGraph` und demonstrieren Sie Ihre Implementierung. Fügen Sie eine Memberfunktion hinzu, die einen Knoten s entgegennimmt und die von s aus erreichbaren Knoten als Array zurückgibt. Dabei soll die Knotenreihenfolge im Rückgabearray bezüglich der Breitensuche ab s entsprechen. Weisen Sie nach, dass Ihr Verfahren auch für Graphen mit gerichteten Kreis korrekt funktioniert.

Um das Projekt zu kompilieren, soll (im Stammverzeichnis des Projekts) folgender Compileraufruf verwendet werden. Alternativ verwenden Sie das Buildsystem `cmake`.

```
clang++ -std=c++17 -I./include -I./external \  
src/simpler_graph.cpp \  
examples/aufg1.cpp -o aufg1
```

Aufgabe 2 (Tiefensuche). In Ihrem Abgabe-Repository hat Flavius einen Branch mit dem Namen `Zettel_07` angelegt. Erweitern Sie die dort, unter Verwendung der Projektstruktur, Ihre Implementierung der Klasse `SimplerGraph` und demonstrieren Sie Ihre Implementierung. Fügen Sie eine Memberfunktion hinzu, die einen Knoten s entgegennimmt und die von s aus erreichbaren Knoten als Array zurückgibt. Dabei soll die Knotenreihenfolge im Rückgabearray bezüglich der Tiefensuche¹ ab s entsprechen. Weisen Sie nach, dass Ihr Verfahren auch für Graphen mit gerichteten Kreis korrekt funktioniert.

Um das Projekt zu kompilieren, soll (im Stammverzeichnis des Projekts) folgender Compileraufruf verwendet werden. Alternativ verwenden Sie das Buildsystem `cmake`.

```
clang++ -std=c++17 -I./include -I./external \  
src/simpler_graph.cpp \  
examples/aufg2.cpp -o aufg2
```

Aufgabe 3 (Elemente aus doppelt verketteten Listen entfernen). Beschreiben Sie, wie ein generelles Element einer doppelt verketteten Liste algorithmisch entfernt wird. Unterscheiden Sie die Fälle, dass sich das Element am Anfang, am Ende, oder weder am Anfang noch am Ende befindet. Ihre Beschreibung darf unterstützende Zeichnungen enthalten.

¹Sie dürfen den Datentyp `std::queue<int>` verwenden. Lesen Sie sich vor der Verwendung eine geeignete Quelle durch, welche die Verwendung ausführlich darstellt.