

Lösung zu Aufgabe 5.7

ALGO1 · SoSe 2021

Michelangelo Battista <s7657928@stud.uni-frankfurt.de>

Noura Ettalibi <s3638690@stud.uni-frankfurt.de>

Fabian Lebert <s5914579@stud.uni-frankfurt.de>

3. Dezember 2021

Aufgabenteil (a)

Pseudocode:

```
1 def Clique(GraphG, NodesAmount):
2     referenceNumber = Blackbox(GraphG)
3     for i in range(0, NodesAmount, 1):
4         tempG = RemoveOneNode(GraphG, i)
5         tempNumber = Blackbox(tempG)
6         if tempNumber == referenceNumber:
7             GraphG = tempG
8     return GraphG
```

Beschreibung des Algorithmus:

Unser Algorithmus besteht aus 2 Funktionen. Die Hauptfunktion "Clique()" braucht 2 Argumente genannt "GraphG" und "NodesAmount".

"GraphG" ist der Graph aus dem ein vollständiger Teilgraph maximaler Größe berechnet werden soll. "NodesAmount" ist ein Integer, indem die Anzahl der Knoten in "GraphG" angegeben sind.

Um die maximale Größe des vollständigen Teilgraphens heraus zu finden, wird in Zeile zwei die Blackbox mit dem gesamten Graphen G aufgerufen. Die maximale Größe wird dann in einer Variable namens "referenceNumber" gespeichert.

Die For-Schleife entspricht nun der Hauptidee des Algorithmuses. Um den vollständigen

Teilgraph von G heraus zu finden wird jetzt jeder Knoten nacheinander aus dem Graphen G gelöscht und in der Variable "tempG" gespeichert. Der somit entstandene neue Graph wird jetzt in die Blackbox gegeben, um erneut die maximale Größe des vollständigen Teilgraphen zu berechnen und in der Variable "temp" gespeichert. Wenn der Wert aus der Variable "temp" gleich dem Wert der Variable "referenceNumber" ist, hat der entfernte Knoten keine Bedeutung in dem vollständigen Teilgraphen. Daraus folgt, dass der Knoten dauerhaft aus dem "GraphG" entfernt werden kann. Die dauerhafte Entfernung findet in Zeile 7 statt. Sobald die For-Schleife fertig durchlaufen ist, wird am Ende der fertige Teilgraph returnt.

Die zweite wichtige Funktion heißt "RemoveOneNode" und ist dafür verantwortlich den Knoten "i" aus dem übergebenen "GraphG" zu entfernen. Wir nehmen dafür an, dass wir eine doppeltverkettete Liste benutzen, die wiederum alle Knoten als doppeltverkettete Listen verwaltet. Somit wird zu Entfernung eines Knoten eine Laufzeit von $O(n^2)$ gebraucht.

Laufzeit und Korrektheit:

Die For-Schleife des Algorithmus läuft in Zeit $\mathcal{O}(n)$. Die Blackbox gibt ihr Ergebnis in polynomieller Zeit aus und RemoveOneNode läuft in Zeit $\mathcal{O}(n^2)$, da wir davon ausgehen, dass unserer Graph als doppelt verkettete Liste gegeben ist, damit läuft unsere for-Schleife in $\mathcal{O}(n^3)$ oder in der Zeit der Blackbox liegt, abhängig davon was größer wächst, jedoch ist beides polynomiell, damit läuft der Algorithmus in polynomieller Laufzeit.

Der Algorithmus bekommt einen Graphen G und die Anzahl der Knoten gegeben, am Anfang wird gespeichert, wie groß der größte vollständige Teilgraph ist, in der for-Schleife werden die Knoten entfernt, welche die Größe des vollständigen Teilgraphen verändern, dadurch sind am Ende nur noch die Knoten vorhanden, welche essenziell für den größten vollständigen Teilgraphen sind.

Aufgabenteil (b)

Aufgabenteil (c)