

Programmieraufgabe 7

Robert Wettstädt 535161
Sona Pecenakova 540607

1. Führen Sie eine Breitensuche BFS mit Startknoten s im unten abgebildeten Graphen durch. Notieren Sie den Verlauf der Zustände der Warteschlange Q .

```
Q = {s}

1. u = s(0)      Q = {a(1), d(1)}
2. u = a(1)      Q = {d(1), b(2), e(2)}
3. u = d(1)      Q = {b(2), e(2), f(2)}
4. u = b(2)      Q = {e(2), f(2), c(3), g(3)}
5. u = e(2)      Q = {f(2), c(3), g(3)}
6. u = f(2)      Q = {c(3), g(3), h(3)}
7. u = c(3)      Q = {g(3), h(3), i(4)}
8. u = g(3)      Q = {h(3), i(4)}
9. u = h(3)      Q = {i(4)}
10. u = i(4)     Q = {}
```

2. Lesen Sie den Beweis für die Korrektheit von BFS auf den Seiten 11 und 12 durch. Wieso gilt 3. auf Seite 30? Wo wird dies im Beweis verwendet?

- Seite 30 Punkt 3 besagt: Zu jedem Zeitpunkt gilt für die Warteschlange, dass die Differenz zwischen $d[\text{Kopfknoten}]$ und $d[\text{letzter Knoten}]$ in der Queue ≤ 1 ist.
- dies gilt, weil zu einem Vertex v immer nur seine direkten Nachbarn untersucht werden
- die Differenz von einem Vertex v zu seinem Nachbarn z beträgt 1
- erst wenn alle Nachbarn von v untersucht wurden, wird die Suche vertieft, doch zu diesem Zeitpunkt wurde v bereits aus der Queue entfernt
- dies ist im Beweis zu finden im Punkt: Wegen 2. ist damit $d[v] \geq d(s, v) > d[z]$. Also wird z vor v aus der Schlange entfernt. Wir betrachten den Zeitpunkt, bei dem z der Kopf der Schlange ist.
- hier wird betont, dass z aus der Queue entfernt wird, bevor v entfernt wird Fall 1: v ist in diesem Moment schon in Q . Dann ist
$$d[v] \leq d[z] + 1 = d(s, z) + 1 = d(s, v)$$
- wenn z Kopf von Q ist und v bereits in Q ist, dann ist die Tiefe von v kleiner gleich der Tiefe von $z + 1$ Fall 2: v wird erst jetzt wegen der Kante zwischen z und v in Q aufgenommen und zwar mit dem Wert
$$d[v] = d[z] + 1 = d(s, v)$$
- v wird als direkter Nachbar von z hinzugefügt, mit einer Tiefe von $z + 1$
- bevor die Nachbarn von v mit einer Tiefe von $v + 1$, bzw. einer Tiefe von $z + 2$
- doch zu dem Zeitpunkt ist z bereits aus der Q entfernt

3. Führen Sie eine Tiefensuche DFS mit Startknoten s im unten abgebildeten Graphen durch. Notieren Sie für jeden Knoten u das Zeitintervall $(d[u], f[u])$ und für jede Kante, ob sie eine Tree-, Back-, Forward- oder Cross-Kante ist. Starten Sie mit dem Knoten s und gehen Sie die Adjazenzlisten lexikographisch durch.

