Aufgabensammlung Teil 7

Thema: Graphdurchmusterung

Aufgabe 1: Pseudobäume

Dies ist eine alte Klausuraufgabe!

Ein **ungerichteter** (zusammenhängender) Baum kann dadurch charakterisiert werden, dass er keine Zykel enthält. Dagegen enthält ein *Pseudobaum* genau einen Zykel. Sei T ein Pseudobaum mit n Knoten.

a) Wie viele Kanten hat T? Begründen Sie Ihre Antwort.

b) Zeigen Sie: Wird eine Kante aus dem Zykel von T gelöscht, so ist der resultierende Graph immer noch zusammenhängend.

c) Sei $G = (V, E)$ ein ungerichteter zusammenhängender Graph **in Adjazenzlistendarstellung** mit $|V| = n$ und $|E| = m$. Entwerfen Sie einen Algorithmus, der feststellt ob G ein Pseudobaum ist. Geben Sie die Idee und die Laufzeit Ihres Algorithmus an.

Aufgabe 2: Erkennung von Zykelgraphen

Dies ist eine alte Klausuraufgabe!

Sei $G = (V, E)$ ein **gerichteter** Graph **in Adjazenzlistendarstellung** mit $|V| = n$ und $|E| = m$. Geben Sie einen Algorithmus in Pseudocode an, der feststellt, ob G ein Zykel der Länge n ist. Benutzen Sie dabei keinerlei Hilfsfunktionen wie MST, DFS, kürzeste Wege, TopSort oder dergleichen.

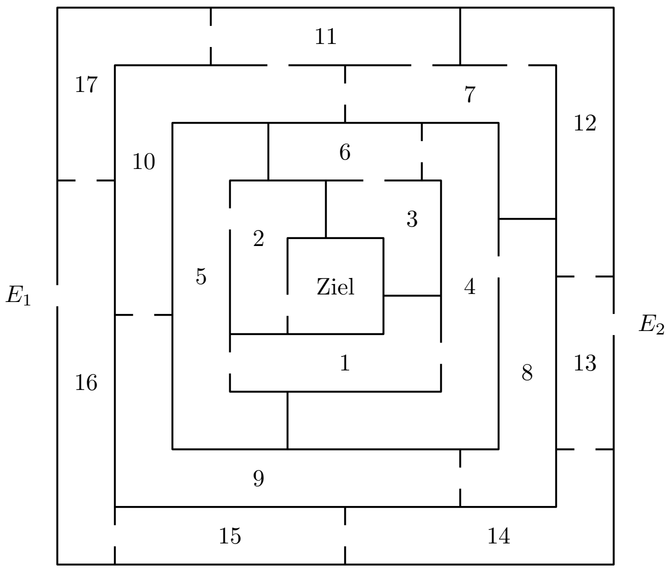
Aufgabe 3: Starker Zusammenhang

Gegeben sei ein gerichteter Graph $G = (V, E)$ mit $n = |V|$ und $m = |E|$. Der Komponentengraph $G' = (V', E')$ ist der Graph, der entsteht, wenn jede starke Zusammenhangskomponente zu einem Knoten geschrumpft wird (vergleiche auch mit den Vorlesungsfolien). Beachten Sie, dass G' ein einfacher Graph ist, d. h. in G' gibt es keine Selbstkanten (v, v) und höchstens eine Kante zwischen jedem Knotenpaar.

Entwerfen Sie einen Algorithmus (nicht in Pseudocode), der den Komponentengraph eines Graphen G in Zeit $\mathcal{O}(n + m)$ berechnet. Begründen Sie die Korrektheit und die Laufzeit Ihres Algorithmus.

Aufgabe 4: Labyrinth

Wie jedes Jahr gibt es auch dieses Mal ein Strohballen-Labyrinth in Stars Hollow. Von oben betrachtet sieht das Labyrinth wie folgt aus:



a) Modellieren Sie das Labyrinth durch einen Graphen $G = (V, E)$. Stellen Sie den Graph dabei mit Hilfe einer Adjazenzliste dar und verwenden Sie die Beschriftungen der Labyrinth-Abschnitte aus der Skizze.

b) Sei ℓ die Anzahl der Labyrinth-Abschnitte und d die Anzahl der Durchgänge. Geben Sie einen Algorithmus an, der bei Eingabe eines (beliebigen) Labyrinths G (repräsentiert als Graph) mit einem Ziel Z und Eingängen $E_1, \ldots, E_k$ ($k \geq 1$) in einer Laufzeit von $\mathcal{O}(\ell + d)$ den kürzesten Weg zum Ziel bestimmt. Die Länge eines Weges ergibt sich dabei durch die Anzahl der durchschrittenen Durchgänge auf diesem Weg. Begründen Sie die Korrektheit und Laufzeit Ihres Algorithmus.

c) Geben Sie einen kürzesten Weg zum Ziel für das Beispiel oben an.

Lösungen:

1. a) $|V|$ Kanten
   b) Beweis durch Widerspruch: Man nehme an, dass der Graph zerfällt, wenn die Kante gelöscht wird. Dann stellt sich heraus, dass der Graph zykelfrei sein muss.
   c) Nutze BFS oder DFS: Es wird exakt eine Nicht-Baum-Kante gefunden. Laufzeit ist $\mathcal{O}(n+m)$.
2. Der Algorithmus nutzt dennoch dasselbe Prinzip wie BFS oder DFS. Ausgehend von einem Knoten s wird immer zu ausgehenden Knoten gegangen. Prüfe dabei, ob der outdegree 1 ist (sonst kein Zykel!). Nach n Schritten erreicht man wieder s.
3. Bestimme die Wurzeln der starken Zusammenhangskomponenten entsprechend der Vorlesung. Verschmelze alle Knoten mit derselben Wurzel und füge eine Kante zwischen zwei enthaltenen Knoten u und v ein genau dann wenn ein Knoten in der SZK u mit einem Knoten in der SZK v verbunden ist.
4. a) V entspricht den Labyrinth-Abschnitten und E den Durchgängen
   b) Wende BFS auf den Graphen an
   c) Lösung: