EA Uebungsblatt 2

Max Springenberg, 177792

2.1 Tiefensuche, Breitensuche, Zusammenhang, starker Zusammenhang: Wiederholung und Verstandnisfragen

2.1.1 Wie ist der Ablauf von Tiefensuche und Breitensuche in ungerichteten Graphen?

In der Tiefensuche wird erst der neuexn Knoten abgearbeitet. In der Breitensuche werden immer erst die neuen Knoten abgearbeitet.

2.1.2 Auf welcher Datenstruktur werden die Algorithmen durchgefuhrt?

Knoten werden in eine Queue zwischengespeichert und in einem Array/ einer Map mit random access markiert.
Verbindungen/ Kanten werden mittels einer Adjazenzliste /-matrix realisiert.
Nummern/ IDs für die Ordnung hinsichtlich der Entdeckung der Knoten in DFS-Liste.

2.1.3 Unter welchen Voraussetzungen ist die Hohe eines Breitensuche-Baumes eindeutig? Gilt bei diesen Voraussetzungen auch, dass der Breitensuche-Baum eindeutig ist?

Die Höhe des Baums ist immer eindeutig, dessen Zusammensetzung jedoch nicht.

2.1.4 Ist fur einen gegebenen Graphen jeder Breitensuche-Baum stets hochstens so tief wie jeder Tiefensuche-Baum?

Ja, da der BFS-Baum generell früher die Knoten abhandelt und dadurch eine gleichmäßigere Aufteilung der Kinder entsteht.

2.1.5 Welche zusatzlichen Kantenklassen gibt es bei Tiefensuche auf gerichteten Graphen?

Baum (T-), Forward (F-), Backward (B-) und Kreuz (C-) Kanten. Sonst gleich.

2.1.6 Wie lassen sich die Kantenklassen im Tiefensuche-Algorithmus unterscheiden?

Baum (T-)	neu/ direkt gefunden
Forward (F-)	Niedriegere auf höhere DFS-Nummer
Backward (B-)	Höhere auf niedrigere DFS-Nummer
Kreuz (C-)	alle anderen Kanten

2.1.7 Wie sind Zusammenhangskomponenten (ZHK) und starke Zusammenhangskomponenten (SZHK) definiert? Worin unterscheiden diese sich?

Existiert eine gerichteter weg von Knoten v nach w mit $v, w \in V$, so schreiben wir $v \to w$.
Starkter Zusammenhang ist definiert als $v \leftrightarrow w \equiv v \to w \land w \to v$

Zusammenhangskomponenten sind für ungerichtete Graphen definiert und besagen, dass man zwischen zwei Knoten hin und her gehen kann.

2.1.8 Welchen Algorithmus haben Sie in der Vorlesung kennen gelernt, um die SZHKs eines Graphen zu berechnen? Wie lautet die asymptotische Laufzeit?

In der Vorlesung wurde der Algorithmus von Ksaraju vorgestellt.
Bei diesem Algorithmus word zwei mal eine DFS durchgeführt. Dabei werden f-Nummern in absteigender Reihenfolge verteilt. Anschließend werden alle Kanten invers gerichtet. Abschließend wird noch eine DFS-Traversierung durchgeführt, ausgehend von f[v] = 1. Die Dabei entstehenden T-Kanten sind starke Zusammenhangskomponenten.
Die Laufzeit beläuft sich auf die der DFS mit $O(|V| + |E|)$

2.2 Starker Zusammenhang Algorithmus durchfuhren

DFS-, f-, ZHK-Nummern

G

Knoten	DFS-Nummer	f-Nummer
A	1	7
B	2	6
C	4	4
D	3	5
E	5	3
F	6	2
G	7	1

G^*

Knoten	DFS-Nummer	ZHK

1. G,E
2. F
3. C,A,B,D

Kantenklassen

G:
$T = \{(A, B), (B, D), (D, C), (E, F), (E, G)\}$
$F = \{(A, D)\}$
$B = \{(G, E)\}$
$C = \{G, F\}$

G^*:
$T = \{(G, E), (C, D), (D, A), (D, B)\}$
$F = \emptyset$
$B = \{(E, G), (A, C)\}$
$C = \{(F, E), (F, G), (B, A), (B, F)\}$

2.3 (Starker) Zusammenhang - Eigenschaften

2.3.1 Sei G ein beliebiger gerichteter Graph. Sind die SZHKs auf G eindeutig?

Die SZHK sind eindeutig, nicht jedoch die Reihenfolge, in der sie gefunden werden.

2.3.2 Sei G ein beliebiger gerichteter Graph. Ergeben sich auf G immer die gleichen SZHKs wie auf G?

Ja, die da sich Zusammenhangskomponenten bei invertierung des Graphens bestehen bleiben.
Beweis:
$A \to B \not\equiv B \to A = A \to^{-1} B$

$A \leftrightarrow B \equiv A \to B \wedge B \to A$
$A \leftrightarrow^{-1} B \equiv B \to A \wedge A \to B \equiv A \leftrightarrow B$
Damit verliert nach Invertierung der Richtungen eine Starke Zusammenhangskomponenten nicht ihre Eigenschaft und ein nicht in der Äquivalenzrelation inbegriffenes Tupel erhält auch nicht die Eigenschaft.

2.3.3 Betrachten Sie den Korrektheitsbeweis des Algorithmus von Kosaraju. Bei dem DFS-Durchlauf auf G kann es vorkommen, dass eine Kante von T_x nach T_{x+1} verlauft. Welcher Kantenklasse werden diese Kanten zugeordnet?

2.4 Beschreiben Sie einen Algorithmus, der das bipartite Matchingproblem mit Hilfe eines Flussalgorithmus lost. Welche Laufzeit hat Ihr Algorithmus?

In der Vorlesung wurde ein Algorithmus vorgestallt, bei dem ähnlich, wie bei Hopfield und Karp eine Quelle und Senke angefügt wurde. Zudem wurden alle Knoten der bipartiten Mengen je nach Mengenzugehörigkeit mit Quelle oder Senke Verbunden und Kanten zur Senke gerichtet.
Jeder Knoten hat die Flusskapazität 1. Es wird sein maximaler Fluss mittels dem Algorithmus von Ford und Fulkerson berechnet.
Die Laufzeit beträgt dabei $O(B(|V| + |E|))$ und ist echt pseudopolynomiell.