Algorithmen und Berechnungskomplexität I WS 15/16

Universität Bonn, Institut für Informatik, Abteilung I

10. Aufgabenblatt zur Vorlesung **Abgabe: 19.01.** (12^{30})

Aufgabe 37: Tiefensuche (4 Punkte)

Betrachten Sie den ungerichteten Graphen $G = (V, E)$ mit Knotenmenge $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ und Adjazenzliste A:

$$\begin{aligned}
v_1 &\rightarrow v_3, v_4, v_5, v_6 \\
v_2 &\rightarrow v_5, v_4 \\
v_3 &\rightarrow v_5, v_1, v_4 \\
v_4 &\rightarrow v_2, v_3, v_1 \\
v_5 &\rightarrow v_3, v_1, v_2 \\
v_6 &\rightarrow v_1
\end{aligned}$$

Zeichnen Sie den Graphen auf und führen Sie in Ihrer Zeichnung $DFS(A, v_1)$ aus. Zählen Sie die Schleifendurchläufe mit und notieren sie in jedem Knoten v_i zwei Werte, nämlich die Nummer des Durchlaufs bei welchem v_i auf den Stack gelegt bzw. vom Stack genommen wurde. Vermerken Sie außerdem neben v_i an der entsprechenden Kante die Durchlaufnummer in der sie aus der Nachbarschaftsliste gelöscht wurde.

Wie viele Schleifendurchläufe haben Sie benötigt? Wie groß ist diese Zahl für einen beliebigen (ungerichteten und zusammenhängenden) Graphen und warum? Welcher Knoten verlässt in einem solchen Graphen den Stack zuletzt?

Bitte wenden!

Aufgabe 38: Graphtraversierung (4 Punkte)

Wir betrachten das am Anfang der Vorlesung vorgestellte Problem der *Graphtraversierung*. Gegeben ist ein zusammenhängender Graph $G = (V, E)$ mit Kanten- und Knotengewichten, und ein Startknoten $v_s \in V$. Das Gewicht eines Knotens v ist w_v und das einer Kante e ist w_e, alle Gewichte sind ≥ 1. Alle Agenten starten auf dem Startknoten v_s, und diese können sich entlang den Kanten von G bewegen. Eine Kante e kann jeweils nur von mindestens w_e Agenten überquert werden, und ein Knoten v darf erstmals nur mit mindestens w_v vielen Agenten betreten werden. Wird v erstmals betreten so werden sofort w_v Agenten auf v platziert – diese dürfen v nicht mehr verlassen. Das Ziel ist mit möglichst wenigen Agenten auf jedem Knoten v mindestens w_v Agenten zu platzieren.

Wir betrachten folgenden Algorithmus zum Traversieren eines Graphen G; der Algorithmus $\textsc{TraversiereBaumOptimal}(T, v_s)$ wird als Black-Box verwendet; sie berechnet eine Strategie mit der Baum T von Knoten v_s aus mit einer kleinstmöglichen Anzahl Agenten traversiert wird.

Algorithmus 1 $\textsc{Graphtraversierung}\text{MST}(G = (V, E), v_s)$

Konstruiere einen minimalen Spannbaum $T = (V, E_T)$ von G;
TraversierungsStrategie $S := \textsc{TraversiereBaumOptimal}(T, v_s)$
return S.

Sei $N := \sum_{v \in V} w_v$ und $w_{max} = \max_{e \in E_T} w_e$. Zeigen Sie

a) S benötigt maximal $N + w_{max}$ Agenten zum Traversieren von G.

b) Zum Traversieren von G benötigt man mindestens w_{max} Agenten.

c) S benötigt zum Traversieren von G maximal doppelt so viele Agenten wie mindestens notwendig.

Es darf angenommen werden, dass alle nicht platzierten Agenten sich immer auf einem Knoten befinden bzw. sich in einem Schritt alle über die gleiche Kante bewegen.

Bitte wenden!

Aufgabe 39: Algorithmen analysieren (4 Punkte)

Betrachten Sie folgende Variation des Algorithmus von Kruskal. Sei $G = (V, E)$ ein zusammenhängender Graph mit positiven Kantengewichten. V ist die Knotenmenge und E die Kantenmenge von G.

Algorithmus 2 SPANNBAUM($G = (V, E)$)

Wähle beliebigen Knoten $v \in V$;
Definiere einen Graphen $T = (V_T, E_T)$ mit $V_T = v$ und $E_T := \emptyset$;
$V := V \setminus v$;
while $V \neq \emptyset$ **do**
  $E' := (V \times V_T) \cap E$
  Sei e' eine Kante aus E' mit minimalem Gewicht, wobei $e' = (v, w)$ und $v \in V, w \in V_T$;
  $V := V \setminus v$;
  $V_T := V_T \cup v$;
  $E_T := E_T \cup e'$
end while
return T.

Berechnet dieser Algorithmus korrekt einen minimalen Spannbaum von Graph G? Begründen Sie Ihre Antwort.

Bitte wenden!

Aufgabe 40: Algorithmen auf Graphen: Kruskal (4 Punkte)

Gegeben sei der nachfolgend abgebildete Graph G. Bestimmen Sie mit dem Algorithmus von Kruskal einen minimalen Spannbaum für G. Geben Sie dabei alle Zwischenschritte an.