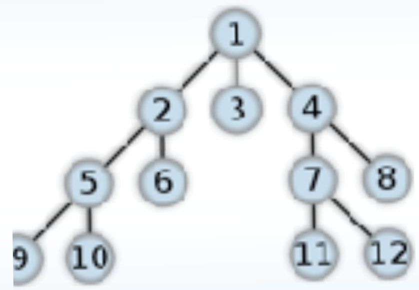

Kapitel 3.3: Zusammenhangskomponenten

Algorithmen und Datenstrukturen
WS 2024/25

Prof. Dr. Sándor Fekete

DEFINITION 3.6 (Wald, Baum)

(1) Ein Wald ist ein kreisfreier Graph.

(2) Ein Baum ist eine Zusammenhangskomponente in einem Wald.
(Also: ein kreisfreier, zusammenhängender Graph)

(3) Ein aufspannender Baum ist ein Baum, der alle Knoten verbindet. (Manchmal auch: Spannbaum. Englisch: "spanning tree")

Algorithmus 3.7

INPUT:

OUTPUT:

Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT:

Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Algorithmus 3.7

<u>INPUT:</u> Graph $G = (V,E)$, Knoten s

<u>OUTPUT:</u> Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$

Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {

Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {

    2.1. Wähle $v \in R$

Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN

Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$

Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {

Algorithmus 3.7

<u>INPUT:</u>	Graph $G = (V,E)$, Knoten s
<u>OUTPUT:</u>	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
    2.1. Wähle $v \in R$
    2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        2.2.1. $R:=R\setminus\{v\}$
    2.3. ELSE {
        2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$

Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R\setminus\{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

Algorithmus 3.7

<u>INPUT:</u> Graph $G = (V,E)$, Knoten s

<u>OUTPUT:</u> Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE $(R \neq \emptyset)$ DO {

    2.1. Wähle $v \in R$

    2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN

        2.2.1. $R:=R \setminus \{v\}$

    2.3. ELSE {

        2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$

        2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

   }

Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
    2.1. Wähle $v \in R$
    2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        2.2.1. $R:=R\setminus\{v\}$
    2.3. ELSE {
        2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
        2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
        }
    }
3. STOP

Satz 3.8. *Der Algorithmus 3.7 ist*

(1) *endlich*

(2) *korrekt.*

Mehr demnächst!

s.fekete@tu-bs.de