

Informatik II

Übung 10

FS 2019

Heutiges Programm

- 1 Wiederholung Vorlesung: Adjazenzlisten
- 2 Breitensuche BFS
- 3 In-Class-Exercise

Adjazenzliste

```
class Graph { // G = (V,E) as adjacency list
    private int V; // number of vertices
    private ArrayList<LinkedList<Integer>> adj; // adj. list
    // Constructor
    public Graph(int n) {
        V = n;
        adj = new ArrayList<LinkedList<Integer>>(V);
        for (int i=0; i<V; ++i)
            adj.add(i,new LinkedList<Integer>());
    }
    // Edge adder method
    public void addEdge(int u,int v) {
        adj.get(u).add(v);
    }
}
```

Adjazenzliste

Eigenschaften:

ArrayList

Elementzugriff in konstanter Zeit.

LinkedList

Element hinzufügen in konstanter Zeit.

Über die ganze Liste iterieren in linearer Zeit.

Adjazenzliste

Eigenschaften:

ArrayList

Elementzugriff in konstanter Zeit.

LinkedList

Element hinzufügen in konstanter Zeit.

Über die ganze Liste iterieren in linearer Zeit.

- $\text{addEdge}(u, v) = \text{adj.get}(u).add(v)$ runs in constant time $\mathcal{O}(1)$.

Adjazenzliste

Eigenschaften:

ArrayList

Elementzugriff in konstanter Zeit.

LinkedList

Element hinzufügen in konstanter Zeit.

Über die ganze Liste iterieren in linearer Zeit.

- `addEdge(u, v) = adj.get(u).add(v)` runs in constant time $\mathcal{O}(1)$.
- `for (int v : adj.get(u))` runs in time $\mathcal{O}(\deg^+(u))$.

Quiz: Laufzeiten einfacher Operationen

Operation	Matrix	Liste
Nachbarn/Nachfolger von $v \in V$ finden		
$v \in V$ ohne Nachbar/Nachfolger finden		
$(u, v) \in E$?		
Kante einfügen		
Kante löschen		

Quiz: Laufzeiten einfacher Operationen

Operation	Matrix	Liste
Nachbarn/Nachfolger von $v \in V$ finden	$\Theta(n)$	
$v \in V$ ohne Nachbar/Nachfolger finden		
$(u, v) \in E$?		
Kante einfügen		
Kante löschen		

Quiz: Laufzeiten einfacher Operationen

Operation	Matrix	Liste
Nachbarn/Nachfolger von $v \in V$ finden	$\Theta(n)$	$\Theta(\deg^+ v)$
$v \in V$ ohne Nachbar/Nachfolger finden		
$(u, v) \in E$?		
Kante einfügen		
Kante löschen		

Quiz: Laufzeiten einfacher Operationen

Operation	Matrix	Liste
Nachbarn/Nachfolger von $v \in V$ finden	$\Theta(n)$	$\Theta(\deg^+ v)$
$v \in V$ ohne Nachbar/Nachfolger finden	$\Theta(n^2)$	
$(u, v) \in E$?		
Kante einfügen		
Kante löschen		

Quiz: Laufzeiten einfacher Operationen

Operation	Matrix	Liste
Nachbarn/Nachfolger von $v \in V$ finden	$\Theta(n)$	$\Theta(\deg^+ v)$
$v \in V$ ohne Nachbar/Nachfolger finden	$\Theta(n^2)$	$\Theta(n)$
$(u, v) \in E$?		
Kante einfügen		
Kante löschen		

Quiz: Laufzeiten einfacher Operationen

Operation	Matrix	Liste
Nachbarn/Nachfolger von $v \in V$ finden	$\Theta(n)$	$\Theta(\deg^+ v)$
$v \in V$ ohne Nachbar/Nachfolger finden	$\Theta(n^2)$	$\Theta(n)$
$(u, v) \in E$?	$\Theta(1)$	
Kante einfügen		
Kante löschen		

Quiz: Laufzeiten einfacher Operationen

Operation	Matrix	Liste
Nachbarn/Nachfolger von $v \in V$ finden	$\Theta(n)$	$\Theta(\deg^+ v)$
$v \in V$ ohne Nachbar/Nachfolger finden	$\Theta(n^2)$	$\Theta(n)$
$(u, v) \in E$?	$\Theta(1)$	$\Theta(\deg^+ v)$
Kante einfügen		
Kante löschen		

Quiz: Laufzeiten einfacher Operationen

Operation	Matrix	Liste
Nachbarn/Nachfolger von $v \in V$ finden	$\Theta(n)$	$\Theta(\deg^+ v)$
$v \in V$ ohne Nachbar/Nachfolger finden	$\Theta(n^2)$	$\Theta(n)$
$(u, v) \in E$?	$\Theta(1)$	$\Theta(\deg^+ v)$
Kante einfügen	$\Theta(1)$	
Kante löschen		

Quiz: Laufzeiten einfacher Operationen

Operation	Matrix	Liste
Nachbarn/Nachfolger von $v \in V$ finden	$\Theta(n)$	$\Theta(\deg^+ v)$
$v \in V$ ohne Nachbar/Nachfolger finden	$\Theta(n^2)$	$\Theta(n)$
$(u, v) \in E$?	$\Theta(1)$	$\Theta(\deg^+ v)$
Kante einfügen	$\Theta(1)$	$\Theta(1)$
Kante löschen		

Quiz: Laufzeiten einfacher Operationen

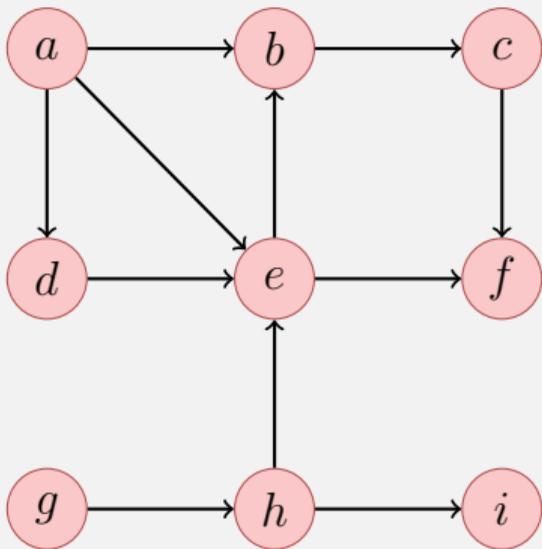
Operation	Matrix	Liste
Nachbarn/Nachfolger von $v \in V$ finden	$\Theta(n)$	$\Theta(\deg^+ v)$
$v \in V$ ohne Nachbar/Nachfolger finden	$\Theta(n^2)$	$\Theta(n)$
$(u, v) \in E$?	$\Theta(1)$	$\Theta(\deg^+ v)$
Kante einfügen	$\Theta(1)$	$\Theta(1)$
Kante löschen	$\Theta(1)$	

Quiz: Laufzeiten einfacher Operationen

Operation	Matrix	Liste
Nachbarn/Nachfolger von $v \in V$ finden	$\Theta(n)$	$\Theta(\deg^+ v)$
$v \in V$ ohne Nachbar/Nachfolger finden	$\Theta(n^2)$	$\Theta(n)$
$(u, v) \in E$?	$\Theta(1)$	$\Theta(\deg^+ v)$
Kante einfügen	$\Theta(1)$	$\Theta(1)$
Kante löschen	$\Theta(1)$	$\Theta(\deg^+ v)$

Breitensuche BFS

BFS von a aus:



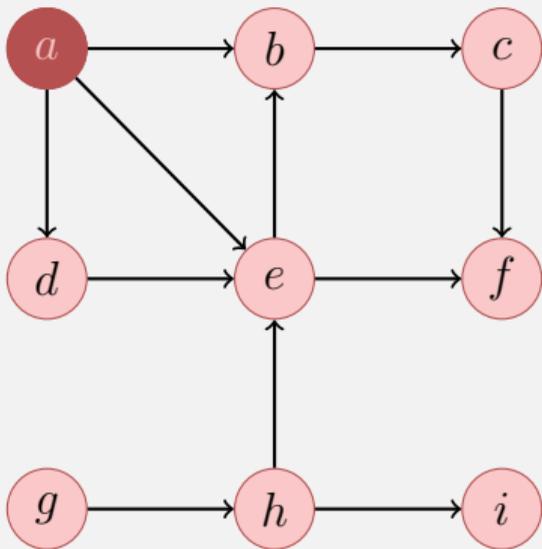
BFS-Baum: Distanzen und Vorgänger



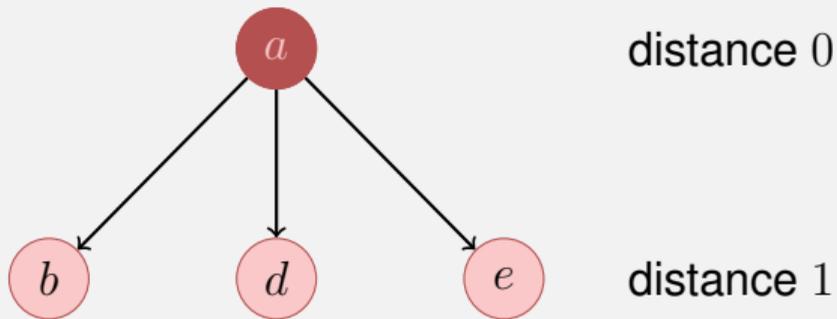
distance 0

Breitensuche BFS

BFS von a aus:

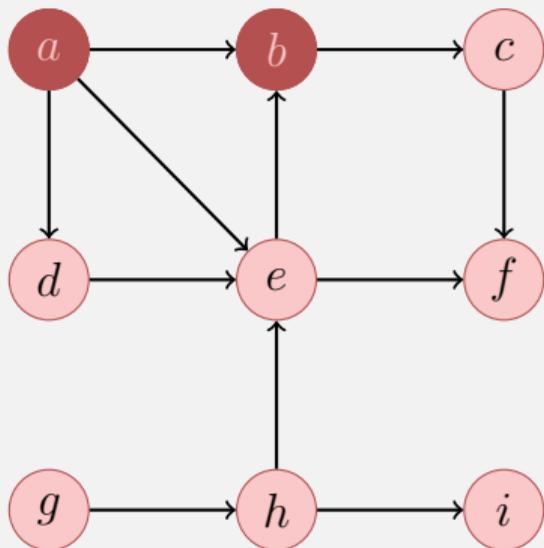


BFS-Baum: Distanzen und Vorgänger

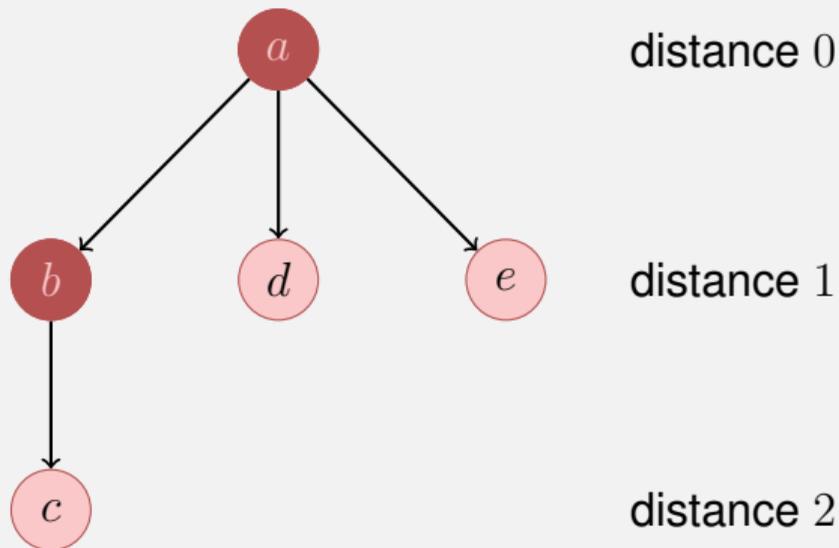


Breitensuche BFS

BFS von a aus:

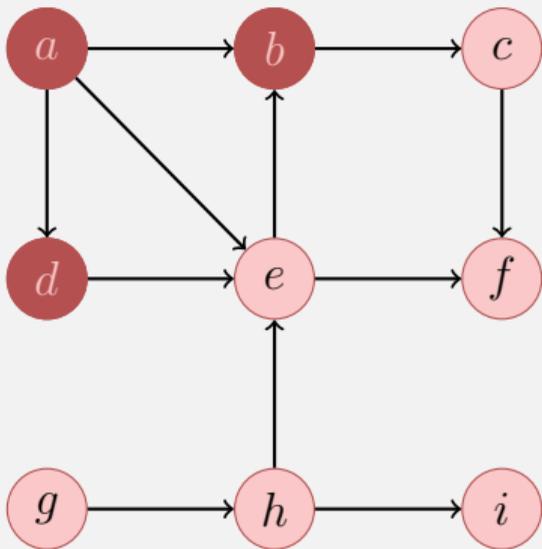


BFS-Baum: Distanzen und Vorgänger

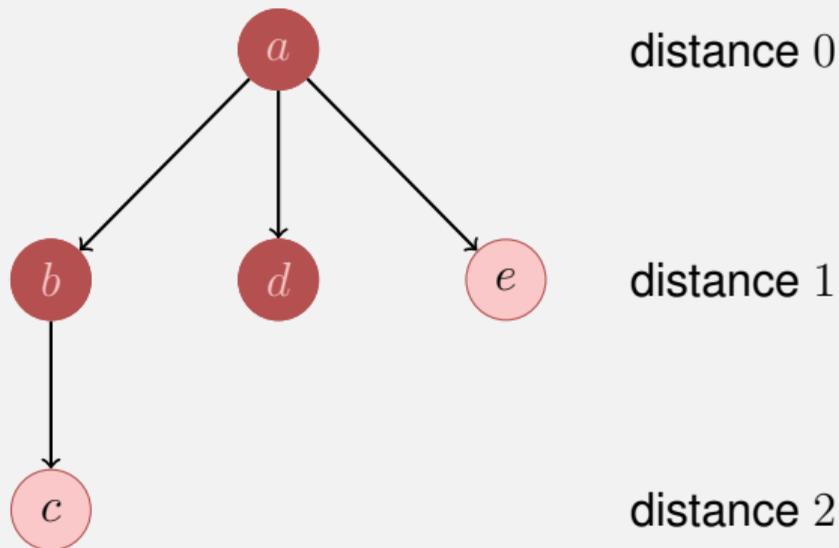


Breitensuche BFS

BFS von a aus:

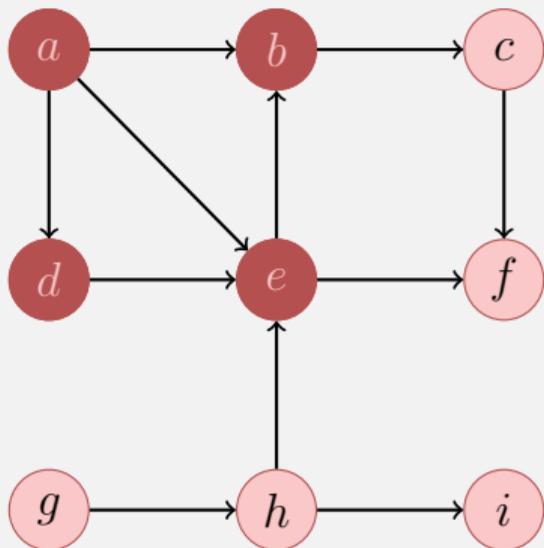


BFS-Baum: Distanzen und Vorgänger

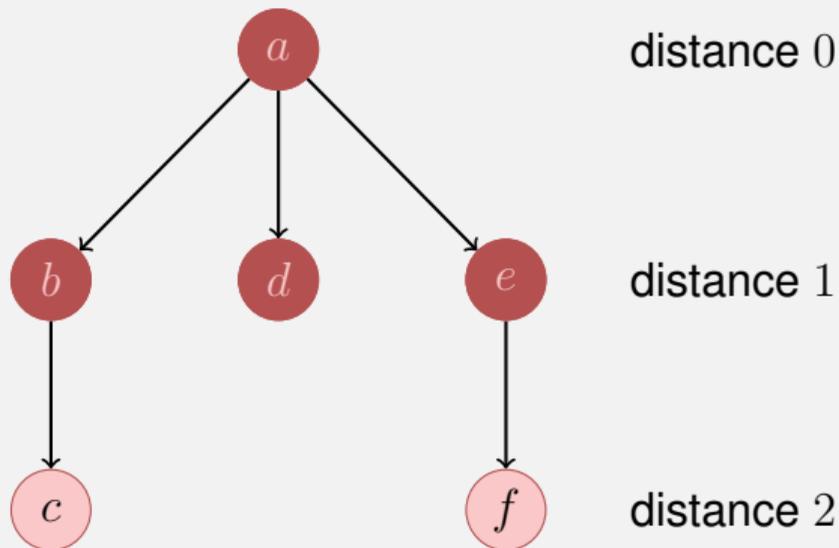


Breitensuche BFS

BFS von a aus:

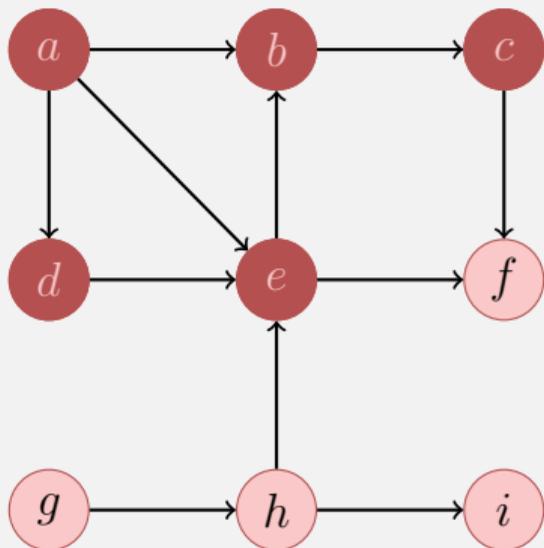


BFS-Baum: Distanzen und Vorgänger

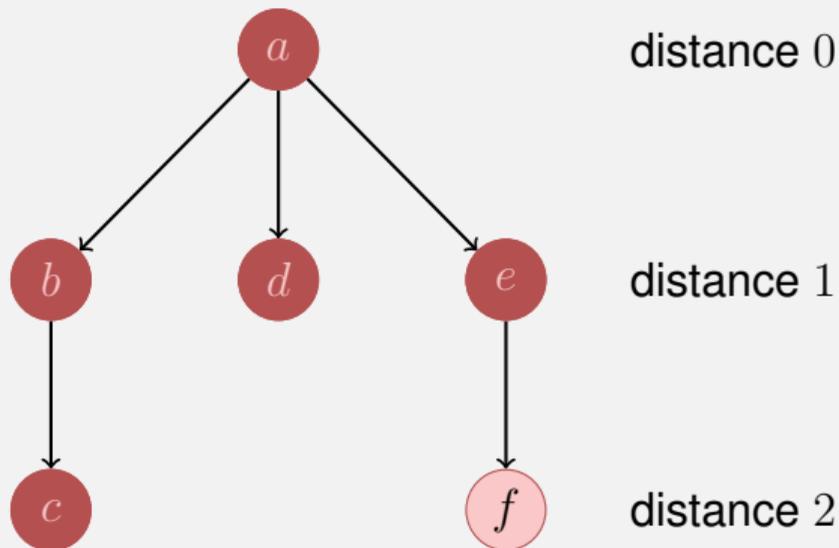


Breitensuche BFS

BFS von a aus:

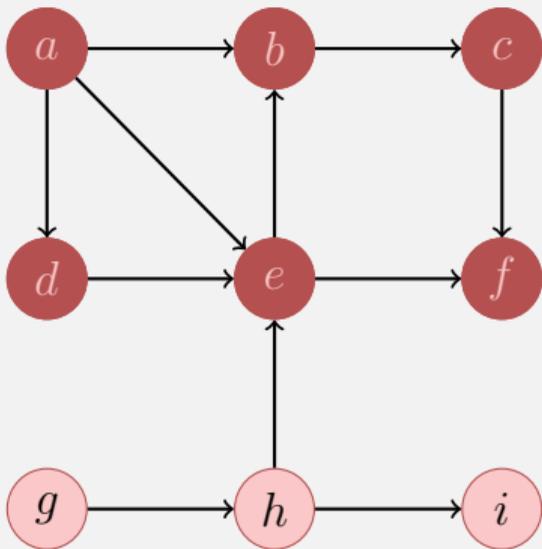


BFS-Baum: Distanzen und Vorgänger

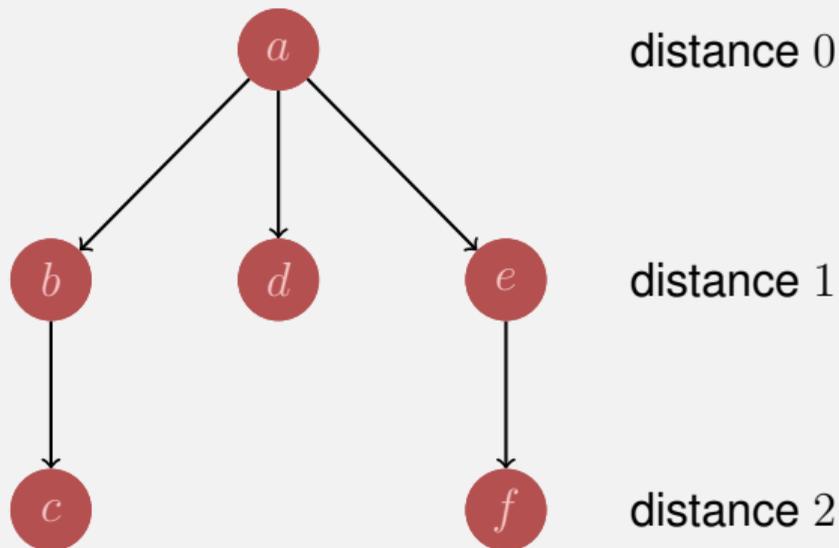


Breitensuche BFS

BFS von a aus:

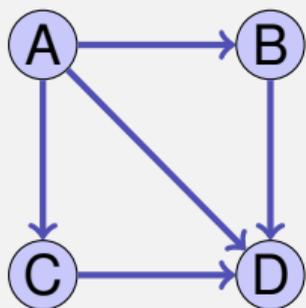


BFS-Baum: Distanzen und Vorgänger

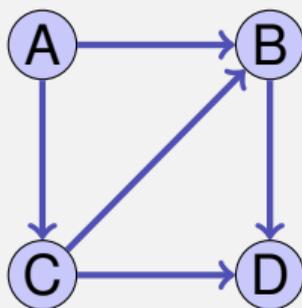


Quiz: Topologisch Sortieren

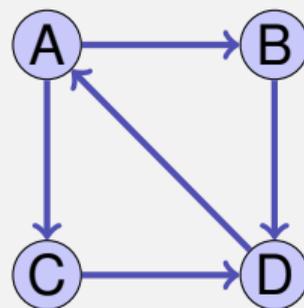
Auf wie viele Arten können die folgenden gerichteten Graphen jeweils topologisch sortiert werden?



Anzahl Sortierungen



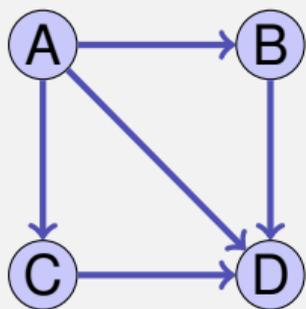
Anzahl Sortierungen



Anzahl Sortierungen

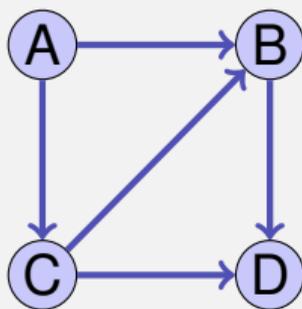
Quiz: Topologisch Sortieren

Auf wie viele Arten können die folgenden gerichteten Graphen jeweils topologisch sortiert werden?



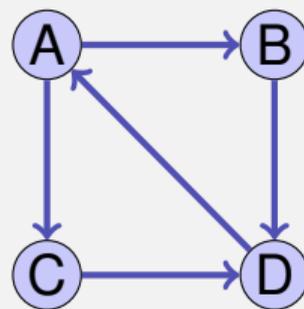
Anzahl Sortierungen

2



Anzahl Sortierungen

1



Anzahl Sortierungen

0

In-Class-Exercises: Wegplanung

Aufgabe: Gegeben sei

- ein gerichteter, ungewichteter Graph $G = (V, E)$, gespeichert als Adjazenzliste,
- und ein spezieller Knoten $t \in V$ (z.Bsp. ein Notausgang).

Entwerfen Sie einen Algorithmus,

- welcher für jeden Knoten $u \in V$ eine Ausgangskante in Richtung eines kürzesten Pfades nach t angibt.
- und eine Laufzeit von $\mathcal{O}(|V| + |E|)$ hat.

In-Class-Exercises: Wegplanung

Lösung:

- 1 Erstelle eine Kopie des Graphen mit umgerichteten Kanten:
 $G^T = (V, E^T)$, mit $E^T = \{(v, u) \mid (u, v) \in E\}$.
Laufzeit: $\mathcal{O}(|V| + |E|)$.
- 2 Starte eine Breitensuche von G^T ausgehend von t ,
und speichere alle BFS-Baum-Kanten.
Laufzeit: $\mathcal{O}(|V| + |E^T|) = \mathcal{O}(|V| + |E|)$.
- 3 Weise die gefundenen Kanten (in umgekehrter Richtung)
den entdeckten Knoten zu.
Laufzeit: $\mathcal{O}(|V|)$.

Fragen oder Anregungen?