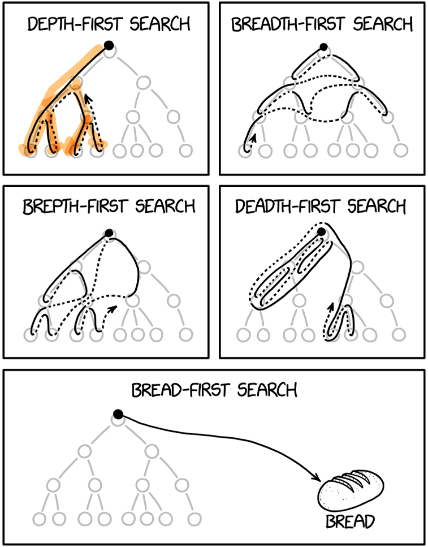
ETH zürich

D&A - Übungsstunde 10

Diese Folien basieren auf denjenigen der Vorlesung, wurden aber durch den Assistenten Adel Gavranović adaptiert und erweitert

Comic der Woche



xkcd

Übersicht

Heutiges Programm

Intro
Follow-up
Feedback zu [code]expert
Wiederholung Graphentheorie
Tipps zu [code]expert
Outro
Live Coding Task

`n.ethz.ch/~agavranovic`

- Link zum Material für die Übungsstunden
- Webseite des Assistenten
- Mail an Assistenten

1. Intro

Intro

Intro

- War letzte Woche krank...

2. Follow-up

Follow-up aus vorherigen Übungsstunden

Follow-up aus vorherigen Übungsstunden

- Wie war’s bei Tanja?

3. Feedback zu [code]expert

Allgemeines zu [code]expert

Allgemeines zu [code]expert

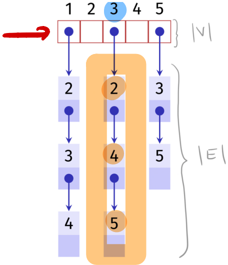
- Nur vereinzelt Fragen im Code

Fragen zu [code]expert eurerseits?

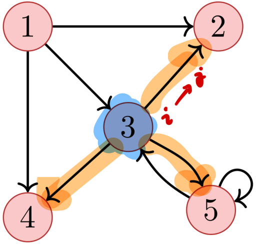
4. Wiederholung Graphentheorie

Matrizen vs. Listen

"Liste"



Graph



"Matrix"

$$
\begin{array}{cc}
 & \begin{array}{ccccc} 1 & 2 & 3 \; j & 4 & 5 \end{array} \\
\begin{array}{c} 1 \\ \\ i \\ \\ \\ \end{array} & \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}
\end{array}
$$

Quiz: Laufzeiten einfacher Operationen

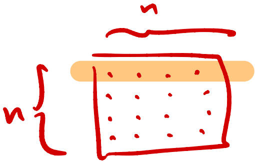
Operation	Matrix	Liste
$(v, u) \in E$?		
Nachbarn/Nachfolger von $v \in V$ finden		
$v \in V$ ohne Nachbar/Nachfolger finden		
alle Kanten $e \in E$ finden		
Kante einfügen		
Kante löschen		

Quiz: Laufzeiten einfacher Operationen

Operation	Matrix	Liste
$(v, u) \in E$?	$\Theta(1)$	
Nachbarn/Nachfolger von $v \in V$ finden		
$v \in V$ ohne Nachbar/Nachfolger finden		
alle Kanten $e \in E$ finden		
Kante einfügen		
Kante löschen		

Quiz: Laufzeiten einfacher Operationen

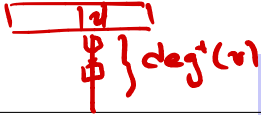
Operation	Matrix	Liste
$(v, u) \in E$?	$\Theta(1)$	$\Theta(\boxed{\deg^+} v)$
Nachbarn/Nachfolger von $v \in V$ finden		
$v \in V$ ohne Nachbar/Nachfolger finden		
alle Kanten $e \in E$ finden		
Kante einfügen		
Kante löschen		



Quiz: Laufzeiten einfacher Operationen

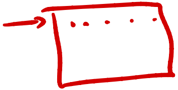
Operation	Matrix	Liste
$(v, u) \in E$?	$\Theta(1)$	$\Theta(\deg^+ v)$
Nachbarn/Nachfolger von $v \in V$ finden	$\Theta(n)$	
$v \in V$ ohne Nachbar/Nachfolger finden		
alle Kanten $e \in E$ finden		
Kante einfügen		
Kante löschen		

Quiz: Laufzeiten einfacher Operationen



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

$\Theta(n)$



Quiz: Laufzeiten einfacher Operationen

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

Quiz: Laufzeiten einfacher Operationen

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

Quiz: Laufzeiten einfacher Operationen

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

Quiz: Laufzeiten einfacher Operationen

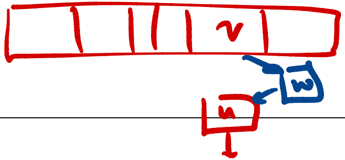
Operation	Matrix	Liste
$(v, u) \in E$?	$\Theta(1)$	$\Theta(\deg^+ v)$
Nachbarn/Nachfolger von $v \in V$ finden	$\Theta(n)$	$\Theta(\deg^+ v)$
$v \in V$ ohne Nachbar/Nachfolger finden	$\Theta(n^2)$	$\Theta(n)$
alle Kanten $e \in E$ finden	$\Theta(n^2)$	$\Theta(n + m)$
Kante einfügen		
Kante löschen		

$m \in O(n^2)$

Quiz: Laufzeiten einfacher Operationen

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

Quiz: Laufzeiten einfacher Operationen



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

Quiz: Laufzeiten einfacher Operationen

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

Quiz: Laufzeiten einfacher Operationen

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

Quiz #1

Frage

Welche Graphdarstellung, Adjazenzmatrix oder Adjazenzliste, eignet sich besser für die Darstellung eines Graphen mit einer hohen Anzahl von Kanten im Vergleich zu Knoten?

Quiz #1

Frage

Welche Graphdarstellung, Adjazenzmatrix oder Adjazenzliste, eignet sich besser für die Darstellung eines Graphen mit einer hohen Anzahl von Kanten im Vergleich zu Knoten?

Antwort

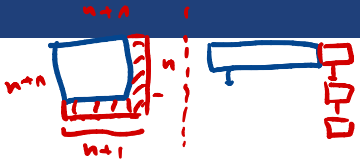
Für einen Graphen mit einer hohen Anzahl von Kanten im Vergleich zu Knoten ist eine Adjazenzmatrix besser geeignet; die Platzkomplexität einer Adjazenzmatrix beträgt $\Theta(n^2)$ und ist unabhängig von der Anzahl der Kanten.

Quiz #2

Frage

Wann wäre es angemessener, eine Adjazenzmatrixdarstellung anstelle einer Adjazenzlistendarstellung zu verwenden? Geben Sie ein Beispiel für ein Szenario.

Quiz #2



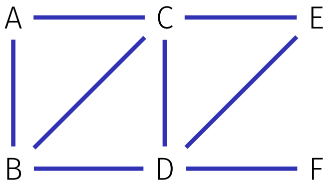
Frage

Wann wäre es angemessener, eine Adjazenzmatrixdarstellung anstelle einer Adjazenzlistendarstellung zu verwenden? Geben Sie ein Beispiel für ein Szenario.

Antwort

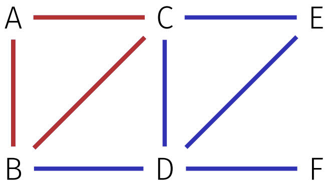
Zum Beispiel in einem Szenario, in dem häufig die Präsenz einer Kante überprüft oder Kanten zwischen Knoten aktualisiert werden müssen, wäre eine Adjazenzmatrix aufgrund ihrer $\Theta(1)$ Kantenabfrage, Einfüge- und Löschzeitkomplexität besser geeignet.

Quiz #3



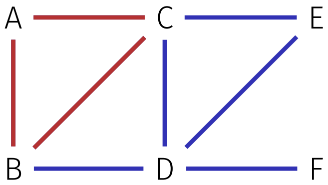
Wir möchten die Anzahl Dreiecke (Kreise mit 3 Knoten und Kanten) in einem Graphen G zählen.

Quiz #3



Wir möchten die Anzahl Dreiecke (Kreise mit 3 Knoten und Kanten) in einem Graphen G zählen.

Quiz #3



Wir möchten die Anzahl Dreiecke (Kreise mit 3 Knoten und Kanten) in einem Graphen G zählen.

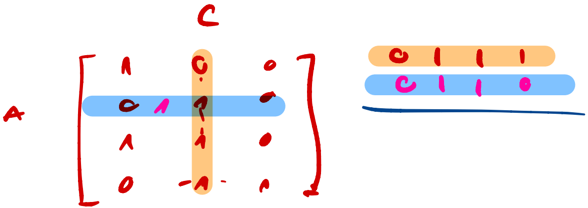
In welcher Zeit können wir das mit einer Adjazenzmatrix tun? Wie sieht es mit einer Adjazenzliste aus?

Quiz #3 Lösung

Adjazenzmatrix:

Quiz #3 Lösung

Adjazenzmatrix: $\Theta(n^2 + m \cdot n)$



Quiz #3 Lösung

Adjazenzmatrix: $\Theta(n^2 + m \cdot n)$

naiv $\Theta(n^3)$: für jede der $\binom{n}{3}$ Kombinationen von 3 Knoten überprüfen, ob die 3 entsprechenden Kanten da sind.

Quiz #3 Lösung

Adjazenzmatrix: $\Theta(n^2 + m \cdot n)$

naiv $\Theta(n^3)$: für jede der $\binom{n}{3}$ Kombinationen von 3 Knoten überprüfen, ob die 3 entsprechenden Kanten da sind.

Effizienter: für jede Kante und jeden zusätzlichen Knoten schauen, ob die zwei zusätzlichen Kanten vorhanden sind.

Quiz #3 Lösung

Adjazenzmatrix: $\Theta(n^2 + m \cdot n)$

naiv $\Theta(n^3)$: für jede der $\binom{n}{3}$ Kombinationen von 3 Knoten überprüfen, ob die 3 entsprechenden Kanten da sind.

Effizienter: für jede Kante und jeden zusätzlichen Knoten schauen, ob die zwei zusätzlichen Kanten vorhanden sind.

Adjazenzliste:

Quiz #3 Lösung

Adjazenzmatrix: $\Theta(n^2 + m \cdot n)$

naiv $\Theta(n^3)$: für jede der $\binom{n}{3}$ Kombinationen von 3 Knoten überprüfen, ob die 3 entsprechenden Kanten da sind.

Effizienter: für jede Kante und jeden zusätzlichen Knoten schauen, ob die zwei zusätzlichen Kanten vorhanden sind.

Adjazenzliste: $\Theta(n \cdot m)$ mit $\Theta(n)$ zusätzlichem Speicher oder $\Theta(n^2 \cdot m)$

Quiz #3 Lösung

Adjazenzmatrix: $\Theta(n^2 + m \cdot n)$

naiv $\Theta(n^3)$: für jede der $\binom{n}{3}$ Kombinationen von 3 Knoten überprüfen, ob die 3 entsprechenden Kanten da sind.

Effizienter: für jede Kante und jeden zusätzlichen Knoten schauen, ob die zwei zusätzlichen Kanten vorhanden sind.

Adjazenzliste: $\Theta(n \cdot m)$ mit $\Theta(n)$ zusätzlichem Speicher oder $\Theta(n^2 \cdot m)$

naiv $\Theta(n^2 \cdot m)$: für jede Kante $e = \{u, v\}$ und jeden potentiellen dritten Knoten w durch die zwei Listen $A[u]$ und $A[v]$ gehen und überprüfen, ob w ein Nachbar ist.

Quiz #3 Lösung

| 1 | 1 | 0 | 1 | 1 | 1 |

Codering für Subsets

| 0 | 1 | 1 | 0 | 1 | 1 |

Adjazenzmatrix: $\Theta(n^2 + m \cdot n)$

naiv $\Theta(n^3)$: für jede der $\binom{n}{3}$ Kombinationen von 3 Knoten überprüfen, ob die 3 entsprechenden Kanten da sind.

Effizienter: für jede Kante und jeden zusätzlichen Knoten schauen, ob die zwei zusätzlichen Kanten vorhanden sind.

Adjazenzliste: $\Theta(n \cdot m)$ mit $\Theta(n)$ zusätzlichem Speicher oder $\Theta(n^2 \cdot m)$

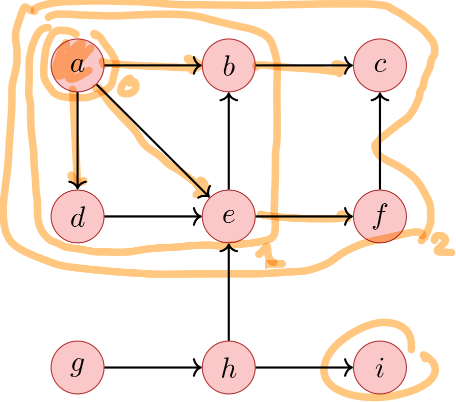
naiv $\Theta(n^2 \cdot m)$: für jede Kante $e = \{u, v\}$ und jeden potentiellen dritten Knoten w durch die zwei Listen $A[u]$ und $A[v]$ gehen und überprüfen, ob w ein Nachbar ist.

Effizienter: einmal durch $A[u]$ gehen, in einem Bitmap der Länge n alle Nachbarn von u abspeichern, dann durch die Nachbarn in $A[v]$ gehen und mit der Bitmap abgleichen.

Breitensuche BFS

a | b d e | c f

BFS von a aus:

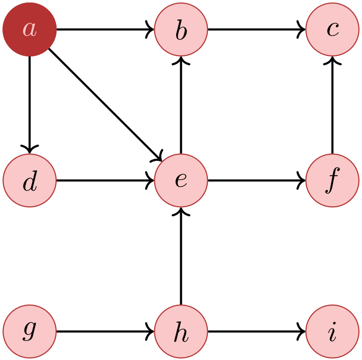


BFS-Baum: Distanzen und Vorgänger

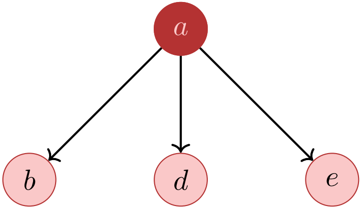
Distanz 0

Breitensuche BFS

BFS von a aus:



BFS-Baum: Distanzen und Vorgänger

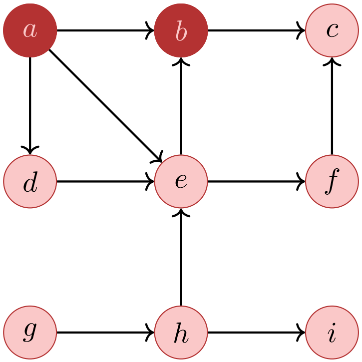


Distanz 0

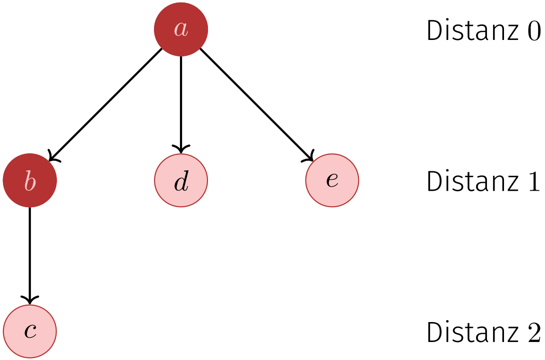
Distanz 1

Breitensuche BFS

BFS von a aus:

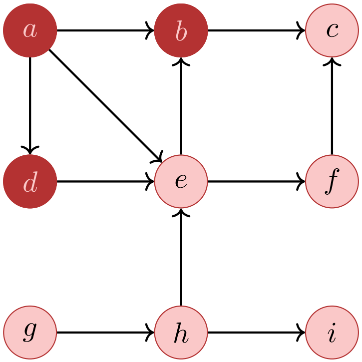


BFS-Baum: Distanzen und Vorgänger

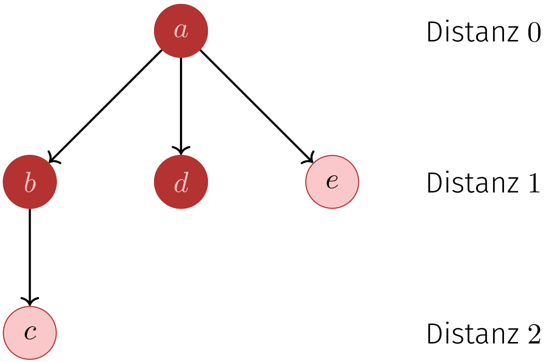


Breitensuche BFS

BFS von a aus:

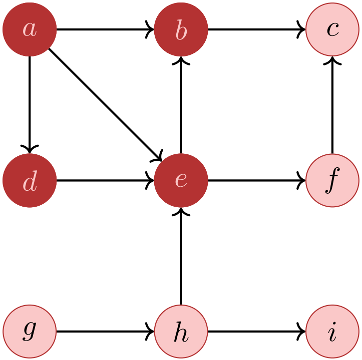


BFS-Baum: Distanzen und Vorgänger

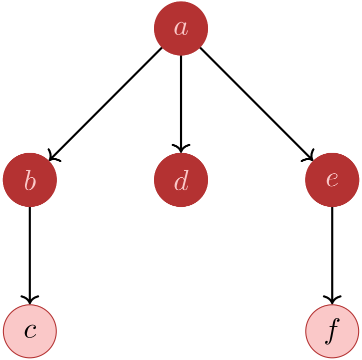


Breitensuche BFS

BFS von a aus:



BFS-Baum: Distanzen und Vorgänger



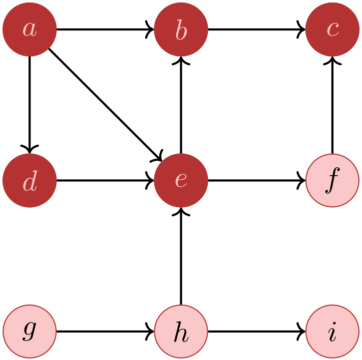
Distanz 0

Distanz 1

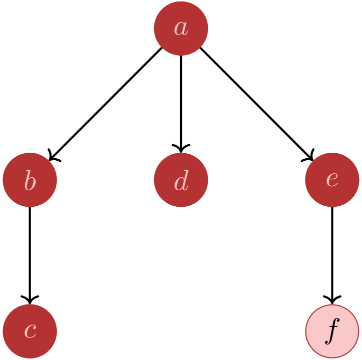
Distanz 2

Breitensuche BFS

BFS von a aus:



BFS-Baum: Distanzen und Vorgänger



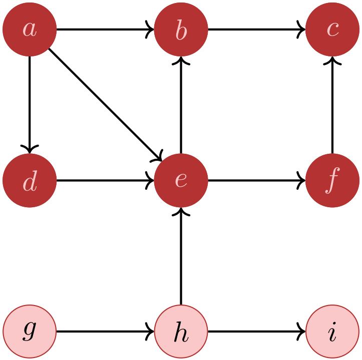
Distanz 0

Distanz 1

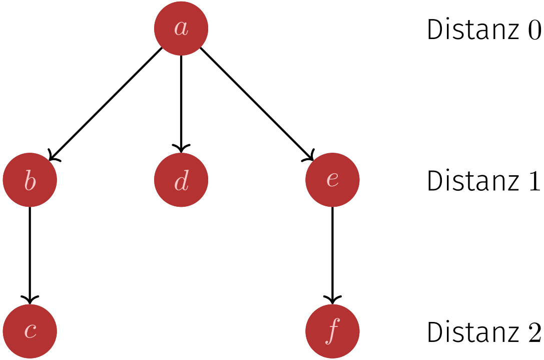
Distanz 2

Breitensuche BFS

BFS von a aus:



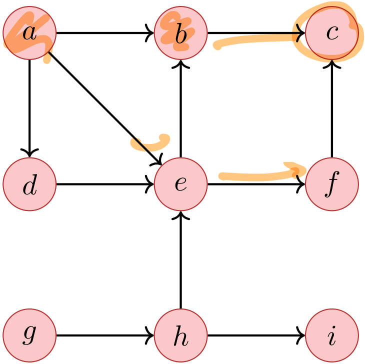
BFS-Baum: Distanzen und Vorgänger



DFS

a, b, ...

DFS von a aus:

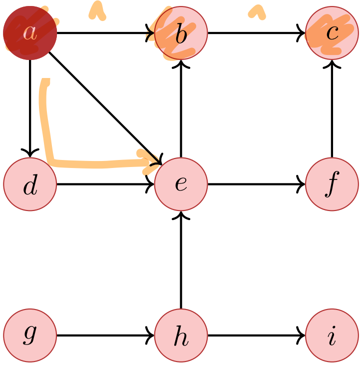


DFS-Baum: Distanzen und Vorgänger

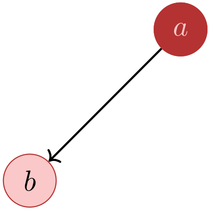
Distanz 0

DFS

DFS von a aus:



DFS-Baum: Distanzen und Vorgänger

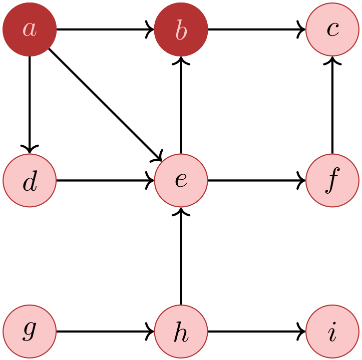


Distanz 0

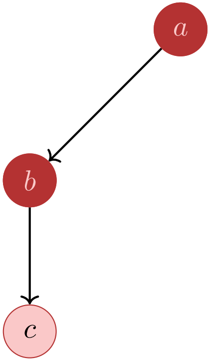
Distanz 1

DFS

DFS von a aus:



DFS-Baum: Distanzen und Vorgänger



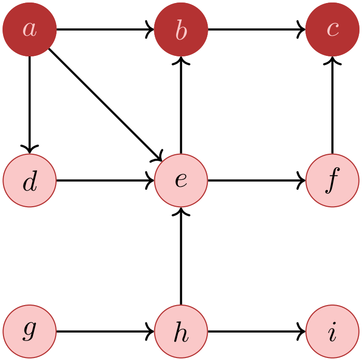
Distanz 0

Distanz 1

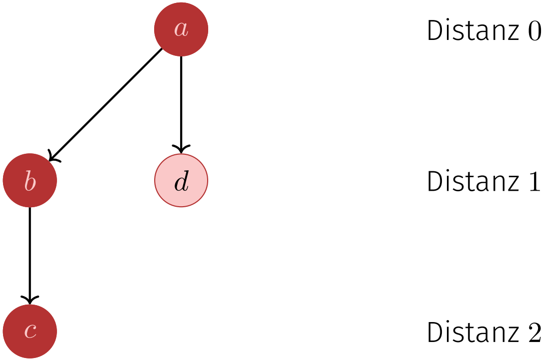
Distanz 2

DFS

DFS von a aus:

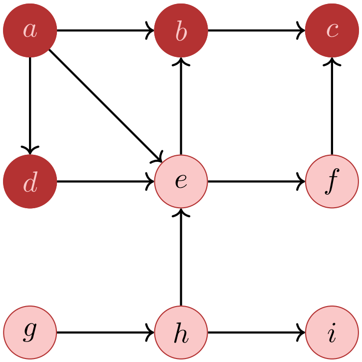


DFS-Baum: Distanzen und Vorgänger

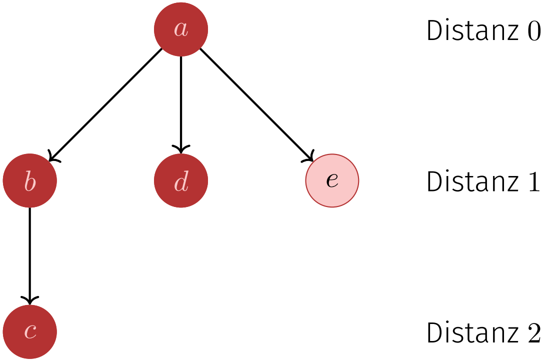


DFS

DFS von a aus:

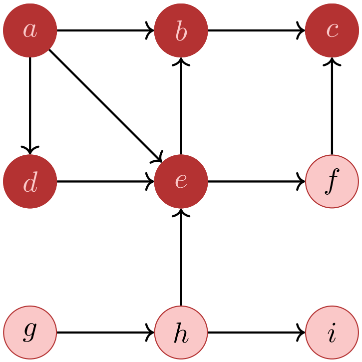


DFS-Baum: Distanzen und Vorgänger

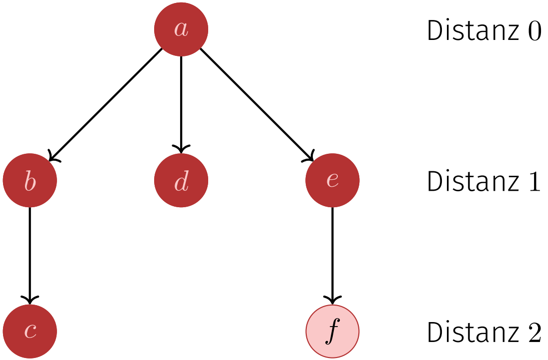


DFS

DFS von a aus:

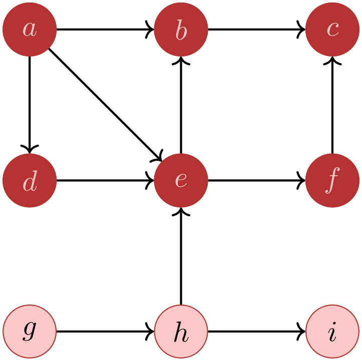


DFS-Baum: Distanzen und Vorgänger

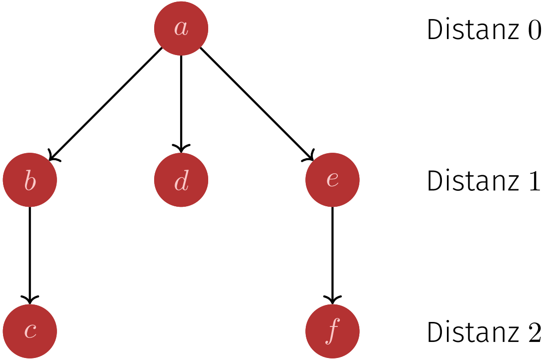


DFS

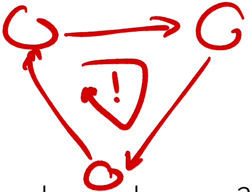
DFS von a aus:



DFS-Baum: Distanzen und Vorgänger



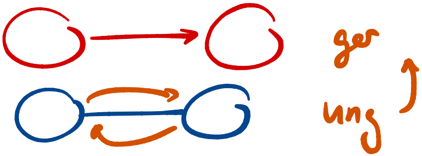
Kreise detektieren



Detektieren von Kreisen

Wie können Sie einen Kreis (cycle) in einem Graphen erkennen? Erklären Sie den Prozess für ungerichtete und gerichtete Graphen.

Kreise detektieren



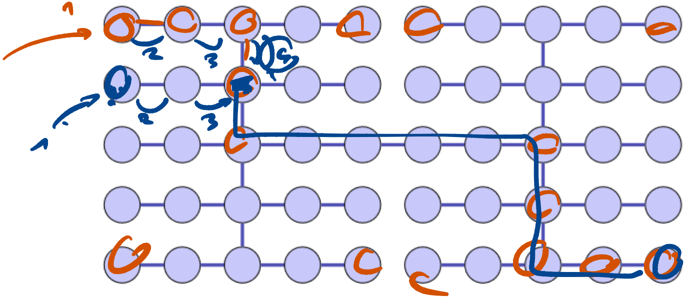
DFS Kreiserkennung

- DFS-Traversierung von einem beliebigen Knoten aus starten
- ungerichtet: Wenn ein besuchter Knoten erneut gefunden wird (mit Ausnahme des unmittelbaren Elternteils), ist ein Zyklus vorhanden
- gerichtet: Wenn eine Kante zu einem grauen Knoten gefunden wird, ist ein gerichteter Kreis vorhanden!

Beispiel Prüfungsaufgabe

Was ist die maximale Rekursionstiefe der (rekursiv implementierten) Funktion DFS angewendet auf folgenden Graphen. Der erste Aufruf wird mitgezählt.

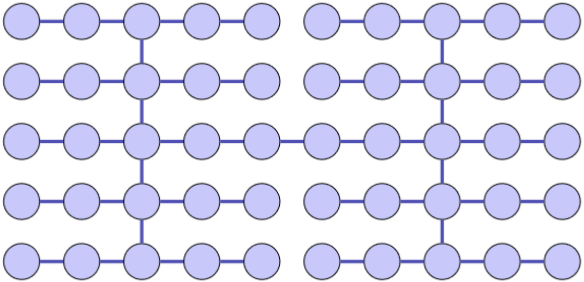
What is the maximum recursion depth of the (recursively implemented) function DFS in the following graph. The first call is counted.



Beispiel Prüfungsaufgabe

Was ist die maximale Rekursionstiefe der (rekursiv implementierten) Funktion DFS angewendet auf folgenden Graphen. Der erste Aufruf wird mitgezählt.

What is the maximum recursion depth of the (recursively implemented) function DFS in the following graph. The first call is counted.

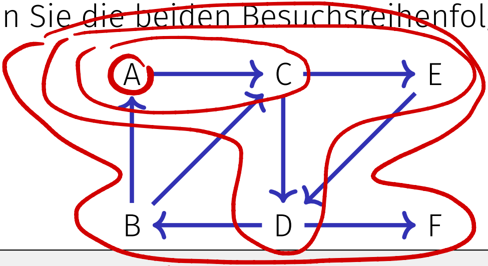


Answer: 14

Fragen/Unklarheiten?

Quiz (von einer alten Prüfung): BFS/DFS

Der folgende Graph wird, ausgehend vom Knoten A aus mit Breitensuche und mit Tiefensuche besucht. Wenn es mehrere Möglichkeiten für eine Besuchsreihenfolge der Nachbarn gibt, wird die alphabetische Ordnung genommen. Geben Sie die beiden Besuchsreihenfolgen an.

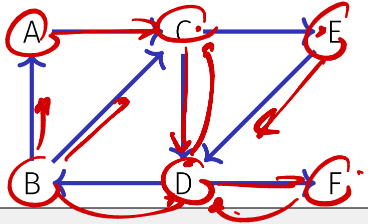


Breitensuche: ? A, C, D, E, B, F

Tiefensuche: ?

Quiz (von einer alten Prüfung): BFS/DFS

Der folgende Graph wird, ausgehend vom Knoten A aus mit Breitensuche und mit Tiefensuche besucht. Wenn es mehrere Möglichkeiten für eine Besuchsreihenfolge der Nachbarn gibt, wird die alphabetische Ordnung genommen. Geben Sie die beiden Besuchsreihenfolgen an.

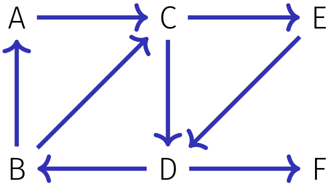


Breitensuche: A C D E B F

Tiefensuche: ? A, C, D, B, F, E

Quiz (von einer alten Prüfung): BFS/DFS

Der folgende Graph wird, ausgehend vom Knoten A aus mit Breitensuche und mit Tiefensuche besucht. Wenn es mehrere Möglichkeiten für eine Besuchsreihenfolge der Nachbarn gibt, wird die alphabetische Ordnung genommen. Geben Sie die beiden Besuchsreihenfolgen an.



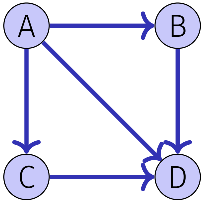
Breitensuche: A C D E B F

Tiefensuche: A C D B F E

Fragen/Unklarheiten?

Quiz 1: Topologisch Sortieren

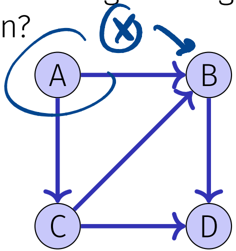
Auf wie viele Arten können die folgenden gerichteten Graphen jeweils topologisch sortiert werden?



Anzahl Sortierungen

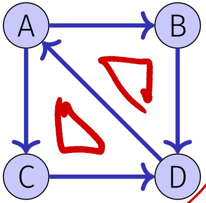
?

A, B, C, D
A, C, B, D } 2



Anzahl Sortierungen

?

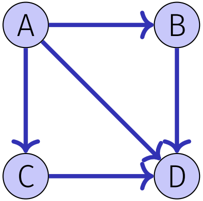


Anzahl Sortierungen

?

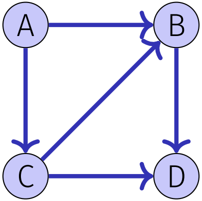
Quiz 1: Topologisch Sortieren

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



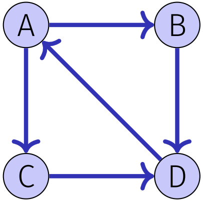
Anzahl Sortierungen

2



Anzahl Sortierungen

1

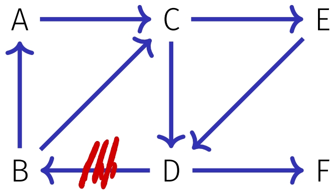


Anzahl Sortierungen

0

Quiz 2: Topologisch Sortieren

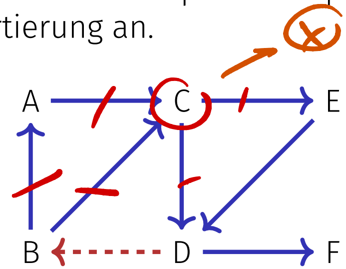
Streichen Sie in folgendem Graphen eine kleinstmögliche Menge von Kanten, so dass der verbleibende Graph eine topologische Sortierung hat. Geben Sie dann eine Sortierung an.



Sortierung: ?

Quiz 2: Topologisch Sortieren

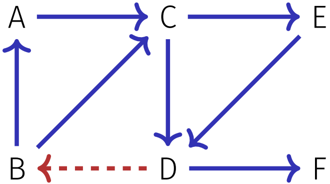
Streichen Sie in folgendem Graphen eine kleinstmögliche Menge von Kanten, so dass der verbleibende Graph eine topologische Sortierung hat. Geben Sie dann eine Sortierung an.



Sortierung: ? B, A, C, E, D, F

Quiz 2: Topologisch Sortieren

Streichen Sie in folgendem Graphen eine kleinstmögliche Menge von Kanten, so dass der verbleibende Graph eine topologische Sortierung hat. Geben Sie dann eine Sortierung an.



Sortierung: B A C E D F

Fragen/Unklarheiten?

5. Tipps zu [code]expert

Tipps für nächste [code]expert -Aufgaben

Aufgabe "Traveling Salesman"

Tipps für nächste [code]expert -Aufgaben

① ② ③ ○ ...

Subset | 0 | 1 | 1 | ...

{2,3}

bit-op

&, |, ^, ~

Aufgabe "Traveling Salesman"

- "Bitstrings" als Mengen verwenden

u int =

| 0 | 0 | 0 | 0 | 1 |

(n)

(n-1), (n) ≟ 1

(n-1)

0
1
10
11
100
101
...

std::vector<bool>

Tipps für nächste [code]expert -Aufgaben

Aufgabe "Traveling Salesman"

- "Bitstrings" als Mengen verwenden
- aber wie?

Aufgabe "Autocorrect"

Tipps für nächste `[code]expert` -Aufgaben

Aufgabe "Traveling Salesman"

- "Bitstrings" als Mengen verwenden
- aber wie?

Aufgabe "Autocorrect"

- Unbedingt Skizzen der DP-Tabellen machen

Aufgabe "Huffmann Code"

Tipps für nächste [code]expert -Aufgaben

Aufgabe "Traveling Salesman"

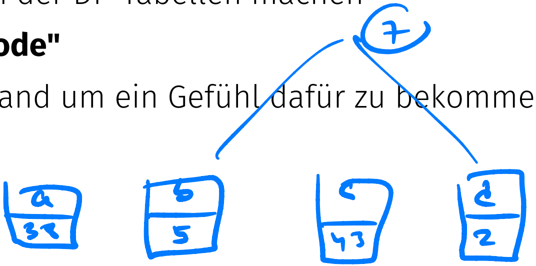
- "Bitstrings" als Mengen verwenden
- aber wie?

Aufgabe "Autocorrect"

- Unbedingt Skizzen der DP-Tabellen machen

Aufgabe "Huffmann Code"

- Macht paar von Hand um ein Gefühl dafür zu bekommen



6. Outro

Allgemeine Fragen?

Bis zum nächsten Mal

Schönes Wochenende!

(bleibt gesund)

7. Live Coding Task

Code-Example 1

’BFS on a Tree’ auf Code-Expert

Code-Example 2

'Sliding Puzzle' auf Code-Expert