Karlsruher Institut für Technologie

Tutorium 8

Algorithmen I SS 14

Institut für Theoretische Informatik

Karlsruher Institut für Technologie

Graphtraversierung

Breitensuche (BFS)

- Funktionsweise:
  1. Beginne bei einem Startknoten s
  2. Besuche alle (unbesuchten) Nachbarknoten von s
  3. Besuche die Nachbarknoten dieser Knoten
  4. ...
- Implementierung über Queue
- Graph wird „Schichtenweise“ durchgegangen (Abstand zum Startknoten)

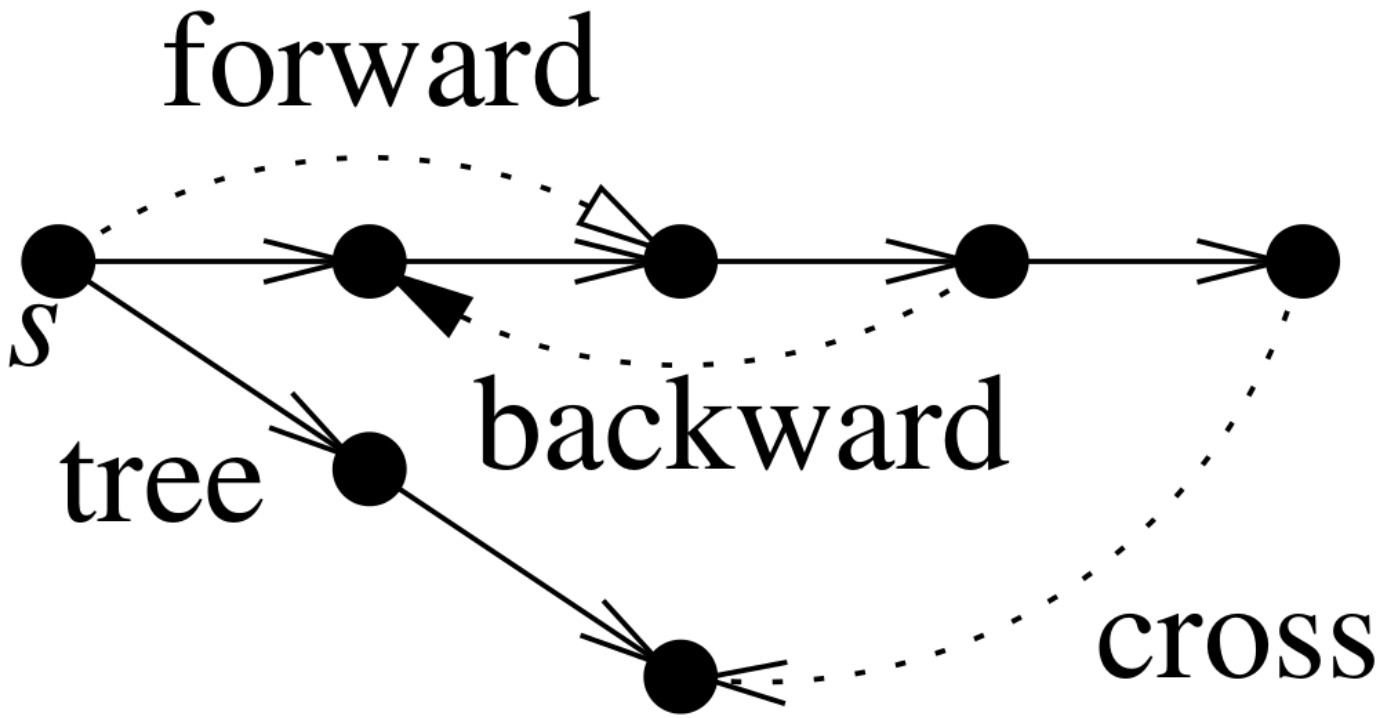
Bipartite Zerlegung

- Ein Graph $G = (V, E)$ heißt bipartit
  $:\Leftrightarrow \exists A, B \subseteq V : A \cap B = \varnothing \wedge \forall (u, v) \in E : (u, v) \in ((A \times B) \cup (B \times A))$
- Also: die Knoten lassen sich in zwei Mengen aufteilen, so dass es keine Kanten innerhalb der Knoten in einer Menge gibt
- Aufgabe: Entwickle einen Algorithmus, der für einen stark zusammenhängenden Graphen eine bipartite Zerlegung findet (oder erkennt, dass keine existiert). Laufzeit: $\mathcal{O}(|E|)$

Tiefensuche (DFS)

- Funktionsweise:
  1. Beginne bei einem Startknoten s
  2. Besuche den ersten (unbesuchten) Nachbarknoten von s
  3. Besuchen den ersten Nachbarknoten dieses Knotens
  4. ...
  5. Besuche den zweiten Nachbarknoten von s
  6. ...
- Intuitiv rekursiv
- Iterative Implementierung?

Kantenklassifizierung



Topologische Sortierung

- Sortiere die Knoten eines Graphen nach der Relation
  $a \leq b :\Leftrightarrow$ Es gibt einen Pfad von a nach b
- Lösung über modifizierte Tiefensuche:
  1. Speichere „Zeitpunkt“ des Knotenabschlusses (keine Nachbarn mehr)
  2. Falls Rückwärtskante gefunden wird $\Rightarrow$ Abbruch
  3. Sortiere die Knoten absteigend nach Abschlusszeit
- Aufgabe: Wie kann Breitensuche modifiziert werden, um eine topologische Sortierung zu berechnen?

Wiederholung

Themen

- O-Kalkül, Mastertheorem
- Invarianten, Korrektheit von Algorithmen
- Verkettete Listen (einfach und doppelt)
- Unbounded Arrays
- Stack, Queue, Deque
- Hashing, Hashtabellen
- Sortieren
  - Insertionsort
  - Mergesort
  - Quicksort (Quickselect)
  - Heapsort
  - Bucketsort
  - Radixsort
- Heaps
- (Binäre) Suchbäume
- Graphen

Aufgaben

Zeigen oder widerlegen Sie, dass $2^{3n} = \mathcal{O}(5^n)$

Aufgaben

Bestimmen Sie die Lösung der folgenden Rekurrenz im Θ-Kalkül mit dem Master-Theorem:

$$T(1) = 5, T(n) = n + 2S(n/2), n = 2^k, k \in \mathbb{N}$$

Aufgaben

Nennen Sie zwei Operationen, die doppelt verkettete Listen in konstanter worst-case-Zeit unterstützen, einfach verkettete Listen jedoch nicht.

Aufgaben

Nennen Sie einen Vorteil und einen Nachteil von einfach verketteten Listen gegenüber unbeschränkten Feldern.

Aufgaben

Wie lautet die Invariante für doppelt verkettete Listen aus der Vorlesung?

Aufgaben

Nennen Sie einen Vorteil und einen Nachteil von Hashing mit verketteten Listen gegenüber offenem Hashing mit linearer Suche.

Aufgaben

Nennen sie einen Vorteil und einen Nachteil von Mergesort gegenüber Quicksort.

Aufgaben

Wie lautet die Heap-Eigenschaft?

Aufgaben

Nennen Sie einen Vorteil und einen Nachteil von Adjazenzfeldern gegenüber Adjazenzlisten.

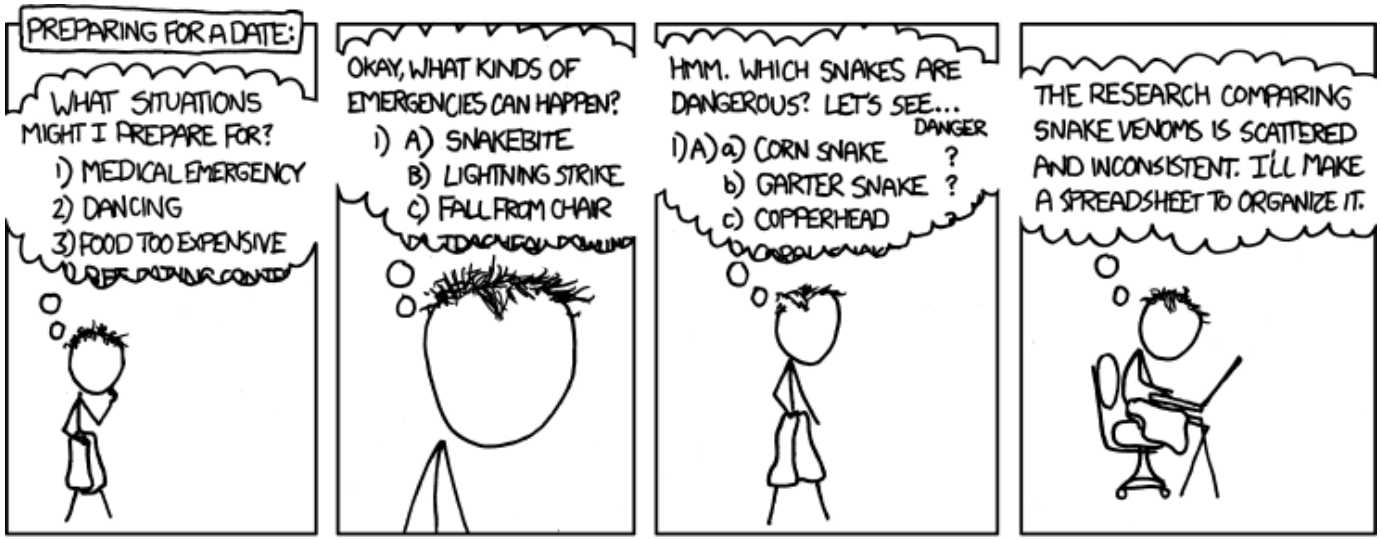
Kreativaufgabe

Gegeben sei ein stark zusammenhänger Graph in folgender Darstellung:

- Knotenarray: Für jeden Knoten ID und Zeiger auf Array mit Kanten
- Kantenarray: Für jede Kante ID der Kante und ID des Zielknotens

Entwirf eine abgewandelte BFS, die *in-place* arbeitet, also nur $\mathcal{O}(1)$ zusätzlichen Platz benötigt. Die Laufzeit darf schlechter sein als eine herkömmliche Breitensuche.

DFS


PREPARING FOR A DATE:
WHAT SITUATIONS MIGHT I PREPARE FOR?
1) MEDICAL EMERGENCY
2) DANCING
3) FOOD TOO EXPENSIVE
OKAY, WHAT KINDS OF EMERGENCIES CAN HAPPEN?
1) A) SNAKEBITE
B) LIGHTNING STRIKE
C) FALL FROM CHAIR
HMM. WHICH SNAKES ARE DANGEROUS? LET'S SEE...
1) A) a) CORN SNAKE
b) GARTER SNAKE
c) COPPERHEAD
DANGER
?
?
THE RESEARCH COMPARING SNAKE VENOMS IS SCATTERED AND INCONSISTENT. I'LL MAKE A SPREADSHEET TO ORGANIZE IT.


I'M HERE TO PICK YOU UP. YOU'RE NOT DRESSED?
BY LD_{50}, THE INLAND TAIPAN HAS THE DEADLIEST VENOM OF ANY SNAKE!
I REALLY NEED TO STOP USING DEPTH-FIRST SEARCHES.