Einsatz und Realisierung von Datenbanksystemen

ERDB Übungsleitung

Alice Rey, Maximilian Bandle, Michael Jungmair

i3erdb@in.tum.de

Folien erstellt von Maximilian Bandle & Alexander Beischl

Organisatorisches
Disclaimer

Die Folien werden von der Übungsleitung allen Tutoren zur Verfügung gestellt.

Sollte es Unstimmigkeiten zu den Vorlesungsfolien von Prof. Kemper geben, so sind die Folien aus der Vorlesung ausschlaggebend.

Falls Ihr einen Fehler oder eine Unstimmigkeit findet, schreibt an i3erdb@in.tum.de mit Angabe der Foliennummer.

Resource Description Framework (RDF)

- Semantisch reichhaltige Beschreibung der Web-Ressourcen
- Nutzt URIs (Uniform Resource Identifiers) um Entities zu identifizieren

- RDF Datenbasis besteht aus:

(Subjekt, Prädikat, Objekt)

- Leicht als Graph zu visualisieren

Resource Description Framework (RDF)

RDF-Beispiel

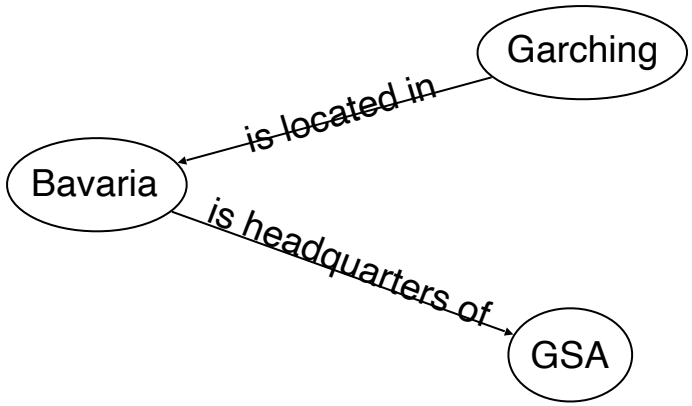
Garching is located in Bavaria.

Subject Predicate Object

Bavaria is headquarters of the German Ski Association.

Subject Predicate Object

Graph representation



SPO triplestore representation

Subject	*Predicate*	*Object*
Garching	is located in	Bavaria
Bavaria	is headquarters of	German Ski Association

Resource Description Framework (RDF)
SPARQL - Anfragesprache für RDF

- Finde alle Personen mit schwarzen Haaren und grünen Augen

```
SELECT ?n
WHERE {
  ?p rdf:type dbo:Person .
  ?p dbo:hairColor "Black" .
  ?p dbo:eyeColor "Green" .
  ?p dbp:name ?n .
}
```

Aufgabe 1

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
<#SOKRATES>
    a foaf:Person ;
    foaf:firstName "Sokrates" ;
    foaf:surName "Sokrates" ;
    foaf:name "Sokrates" ;
    foaf:age 30 ;
    foaf:knows [
        a foaf:Person ;
        foaf:name "Russel"
    ];
    foaf:knows [
        a foaf:Person ;
        foaf:name "MCurie"
    ] .
```

```
<#MCURIE>
    a foaf:Person ;
    foaf:firstName "Marie" ;
    foaf:name "MCurie" ;
    foaf:SurName "Curie" ;
    foaf:age 29 ;
    foaf:knows [
        a foaf:Person ;
        foaf:name "PCurie"
    ];
    foaf:knows [
        a foaf:Person ;
        foaf:name "Russel"
    ];
    foaf:knows [
        a foaf:Person ;
        foaf:name "Sokrates"
    ] .
```

```
<#PCURIE>
    a foaf:Person ;
    foaf:firstName "Pierre" ;
    foaf:name "PCurie" ;
    foaf:SurName "Curie" ;
    foaf:age 29 ;
    foaf:knows [
        a foaf:Person ;
        foaf:name "MCurie"
    ] .
<#RUSSEL>
    a foaf:Person ;
    foaf:firstName "Bertrand" ;
    foaf:name "Russel" ;
    foaf:SurName "Russel" ;
    foaf:age 97 ;
    foaf:knows [
        a foaf:Person ;
        foaf:name "Sokrates"
    ] .
```

Vervollständigen Sie die untere Anfrage um die Namen der Freunde von Personen mit dem Vornamen *Sokrates* zu finden, die älter als 30 Jahre sind. Die *foaf* Onthology is unter `http://xmlns.com/foaf/spec/` beschrieben. Nutzen Sie `https://rdf.db.in.tum.de/` für Ihre Abfrage.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name2
WHERE {
        ....
      }
```

Aufgabe 2

```
@prefix ex: <http://example.org>.
ex:Rapunzel ex:hatAutor ex:Sokrates.
ex:Rapunzel ex:erschienen 2006.
ex:Aschenputtel ex:hatAutor ex:Archimedes.
ex:Aschenputtel ex:hatAutor ex:Platon.
ex:Schneewittchen ex:hatAutor ex:Platon.
ex:Schneewittchen ex:erschienen 2004.
```

Drücken Sie die folgenden Anfragen in SPARQL aus:

1. Geben Sie alle Bücher aus, für die sowohl der Autor als auch das Erscheinungsjahr in der Datenbank enthalten sind.
2. Geben Sie die gemeinsamen Autoren der beiden Bücher Aschenputtel und Schneewitchen aus.
3. Geben Sie die Namen aller Autoren (ohne Duplikate) von Büchern mit einem Erscheinungsjahr nach 2004 aus.

Aufgabe 3

wikidata.org ist ein Projekt, das strukturierte Informationen für Wikimedia-Schwesterprojekte bereitstellt. Informationen über das Datenmodell finden Sie unter https://www.mediawiki.org/wiki/Wikibase/DataModel/Primer. Praktischerweise bietet es auch eine SPARQL-Schnittstelle für Ihre Erkundungen unter query.wikidata.org.

Schreiben Sie SPARQL-Abfragen, um die folgenden Fragen zu beantworten:

1. Listen Sie alles auf, was München als Objekt verwendet. Wikidata hat den URI http://www.wikidata.org/entity/Q1726 für München vergeben. Daher können Sie unter Verwendung einer Präfixdefinition auf München verweisen, indem Sie wd:Q1726 verwenden.
2. Welche Prädikat wird am häufigsten verwendet?
3. Welche der Städte in der Datenbank hat die früheste schriftliche Aufzeichnung?
4. Führen Sie die Unterklassen von Sport (Q349) und ihre Bezeichnungen auf, falls es eine gibt.
5. Listen Sie die transitiven Unterklassen von Sport auf.

Big Data

Big Data
Term Frequency - Inverse Document Frequency

Anwendung im *Information Retrieval*

- Finde zu einer Suchanfrage die relevantesten Dokumente
- Große Herausforderung aufgrund der Menge an Web-Dokumenten

Term Frequency - Inverse Document Frequency (TF-IDF)

- Dokument-Ranking basierend auf Begriffshäufigkeiten
- Vollautomatische Analyse
- Meist wird nur ein *Vokabular* berücksichtigt, nicht alle Worte

Big Data
Term Frequency - Inverse Document Frequency

D_1	D_2	D_3
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$TF_{ij} = \frac{f_{ij}}{\sum_{i=1\ldots|V|} f_{ij}}$$

#Wort i im Dokument j

#Worte im Dokument j

- Gewicht eines Begriffs in einem kurzen Dokument höher als in einem langen Dokument
- Normalisierung

Big Data

Term Frequency - Inverse Document Frequency

D_1	D_2	D_3
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$TF_{ij} = \frac{f_{ij}}{\sum_{i=1\ldots|V|} f_{ij}}$$

#Wort i im Dokument j

#Worte im Dokument j

	TF		
Wort i	D_1	D_2	D_3
ERDB			
Klausur			
Erfolg			

Big Data
Term Frequency - Inverse Document Frequency

D_1	D_2	D_3
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$TF_{ij} = \frac{f_{ij}}{\sum_{i=1\ldots|V|} f_{ij}}$$

#Wort i im Dokument j

#Worte im Dokument j

	TF		
Wort i	D_1	D_2	D_3
ERDB	1/5		
Klausur	0/5		
Erfolg	0/5		

Big Data
Term Frequency - Inverse Document Frequency

D_1	D_2	D_3
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$TF_{ij} = \frac{f_{ij}}{\sum_{i=1\ldots|V|} f_{ij}}$$

#Wort i im Dokument j

#Worte im Dokument j

	TF		
Wort i	D_1	D_2	D_3
ERDB	1/5	0/5	
Klausur	0/5	1/5	
Erfolg	0/5	0/5	

Big Data
Term Frequency - Inverse Document Frequency

D_1	D_2	D_3
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$TF_{ij} = \frac{f_{ij}}{\sum_{i=1\ldots|V|} f_{ij}}$$

#Wort i im Dokument j

#Worte im Dokument j

	TF		
Wort i	D_1	D_2	D_3
ERDB	1/5	0/5	1/10
Klausur	0/5	1/5	1/10
Erfolg	0/5	0/5	1/10

Big Data
Term Frequency - Inverse Document Frequency

D_1	D_2	D_3
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$IDF_i = log(N/n_i)$$

#Dokumente

#Dokumente mit Wort i

- Gewichtung für jeden Begriff
- Seltene Begriffe bekommen eine höhere Gewichtung als Häufige

Big Data
Term Frequency - Inverse Document Frequency

D_1	D_2	D_3
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$IDF_i = log(N/n_i)$$

#Dokumente

#Dokumente mit Wort i

	IDF			
Wort i	**N**	**n_i**	**N/n_i**	**$log(N/n_i)$**
ERDB				
Klausur				
Erfolg				

Big Data
Term Frequency - Inverse Document Frequency

D_1	D_2	D_3
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$IDF_i = log(N/n_i)$$

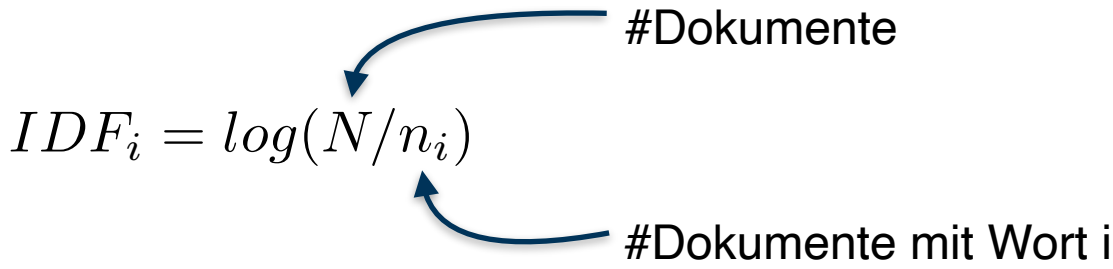
#Dokumente

#Dokumente mit Wort i

	IDF			
Wort i	**N**	**n_i**	**N/n_i**	**$log(N/n_i)$**
ERDB	3			
Klausur	3			
Erfolg	3			

Big Data
Term Frequency - Inverse Document Frequency

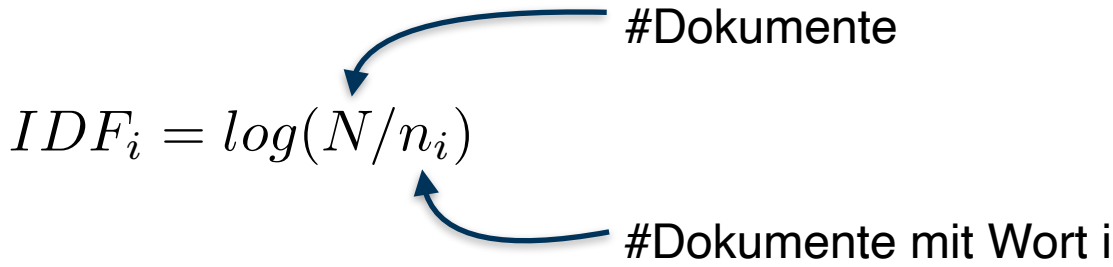
D_1	D_2	D_3
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur



	IDF			
Wort i	**N**	**n_i**	**N/n_i**	**$log(N/n_i)$**
ERDB	3	2		
Klausur	3	2		
Erfolg	3	1		

Big Data
Term Frequency - Inverse Document Frequency

D_1	D_2	D_3
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur



	IDF			
Wort i	**N**	**n_i**	**N/n_i**	**$log(N/n_i)$**
ERDB	3	2	3/2	
Klausur	3	2	3/2	
Erfolg	3	1	3/1	

Big Data
Term Frequency - Inverse Document Frequency

D_1	D_2	D_3
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$IDF_i = log(N/n_i)$$

#Dokumente

#Dokumente mit Wort i

	IDF			
Wort i	**N**	**n_i**	**N/n_i**	**$log(N/n_i)$**
ERDB	3	2	3/2	0,176
Klausur	3	2	3/2	0,176
Erfolg	3	1	3/1	0,477

Big Data

Term Frequency - Inverse Document Frequency

D_1	D_2	D_3
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

Relevanz von Dokument j

$$rel(D_j, Q) = \sum_{i \in Q} TF_{ij} \times IDF_i$$

Anfrage Q

Für alle Wörter von Q

Big Data
Term Frequency - Inverse Document Frequency

D_1	D_2	D_3
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

Relevanz von Dokument j

$$rel(D_j, Q) = \sum_{i \in Q} TF_{ij} \times IDF_i$$

Anfrage Q

Für alle Wörter von Q

rel(D_1,{ERDB,Klausur,Erfolg}) =

$TF_{ERDB,1}$ * IDF_{ERDB} + $TF_{Klausur,1}$ * $IDF_{Klausur}$ + $TF_{Erfolg,1}$ * IDF_{Erfolg}

Big Data
Term Frequency - Inverse Document Frequency

D_1	D_2	D_3
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

Relevanz von Dokument j

$$rel(D_j, Q) = \sum_{i \in Q} TF_{ij} \times IDF_i$$

Anfrage Q

Für alle Wörter von Q

rel(D_1,{ERDB,Klausur,Erfolg}) =

$TF_{ERDB,1}$ * IDF_{ERDB} + $TF_{Klausur,1}$ * $IDF_{Klausur}$ + $TF_{Erfolg,1}$ * IDF_{Erfolg}

rel(D_1,{ERDB,Klausur,Erfolg}) =

1/5 * 0,176 + 0 * 0,176 + 0 * 0,477

Big Data
Term Frequency - Inverse Document Frequency

D_1	D_2	D_3
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

Relevanz von Dokument j

$$rel(D_j, Q) = \sum_{i \in Q} TF_{ij} \times IDF_i$$

Anfrage Q

Für alle Wörter von Q

rel(D_1,{ERDB,Klausur,Erfolg}) =

$TF_{ERDB,1}$ * IDF_{ERDB} + $TF_{Klausur,1}$ * $IDF_{Klausur}$ + $TF_{Erfolg,1}$ * IDF_{Erfolg}

rel(D_1,{ERDB,Klausur,Erfolg}) =

1/5 * 0,176 + 0 * 0,176 + 0 * 0,477 = **0,0352**

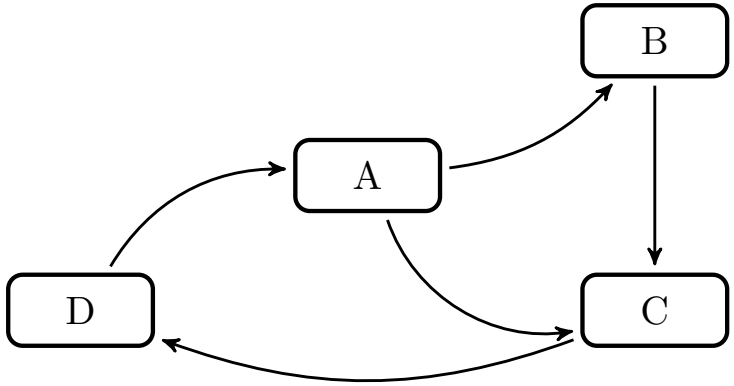
Aufgabe 4

Berechnen Sie für folgende drei Dokumente die TF-IDF-Werte:

1. „Beim Fußball dauert ein Spiel neunzig Minuten – und am Ende gewinnen die Deutschen“
2. „Beim Fußball muss das Runde (der Ball) in das Eckige (das Tor)“
3. „Nie war ein Tor so wertvoll wie jetzt“

Welches Ranking ergibt sich gemäß der Relevanzwerte für die Anfrage: „Fußball“ $\wedge$ „Tor“. Zur Ermittlung des TF Wertes gehen sie davon aus, dass alle Wörter eines Dokuments *interessant* sind?

Aufgabe 5



In dem in Abbildung 1 gezeigten Netzwerk von Web-Seiten wird ein kleines Beispiel für einen Webgraphen gezeigt. Lösen Sie folgende Aufgaben.

1. Berechnen Sie, für das in Abbildung gezeigte Netzwerk, den PageRank, sowie die HITS-Werte nach 2 Iterationen. Nutzen Sie $1/|V|$ als Anfangswert für den PageRank und 1 für HITS. $a = 0.1$
2. Formulieren sie eine Iteration des Pagerank Algorithmus in SQL. Der Graph ist dabei in der Tabelle *edges(src,dst)* gespeichert, die aktuelle PageRank Gewichtung in der Tabelle *pagerank(node,pr)*.
3. Formulieren Sie die SQL Anfrage nun als rekursive SQL Anfrage (100 Iterationen) um.

HITS Algorithmus

Hypertext Induced Topic Selection

Automatische Relevanz-Beurteilung für Websites

Vernetzung als Kriterium

Zwei Rollen:

- Hub (Knotenpunkt)
- Autorität (Website mit Inhalt)

➡ Alle Seiten werden in beiden Rollen beurteilt

Hub

- Wertvoller auf je mehr höherwertige Autoritäten er verweist (ausgehende Kanten)

Autorität

- Wertvoller je mehr höherwertige Hubs auf sie verweisen (eingehende Kanten)

HITS Algorithmus

Hypertext Induced Topic Selection

Iteration:

1. Berechne alle Hub-Werte

Summe der Gewichte der Knoten aller ausgehenden Kanten

$$h_i = \sum_{j=1\ldots N} A_{ij} a_j$$

2. Berechne alle Autoritäts-Werte

Summe der Gewichte der Knoten aller eingehenden Kanten

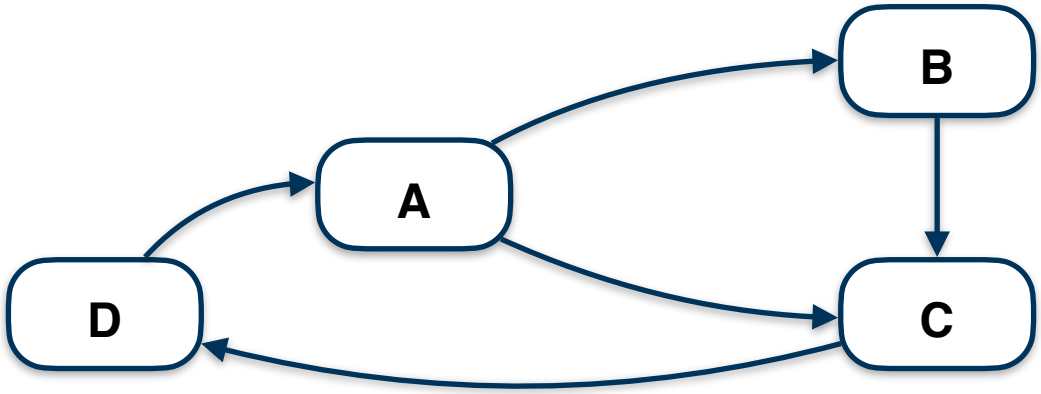
$$a_i = \sum_{j=1\ldots N} A_{ji} h_j$$

3. Normalisiere die Autoritäts-Werte mit

$$\lambda = \frac{1}{max(a)}$$

Aufgabe 2

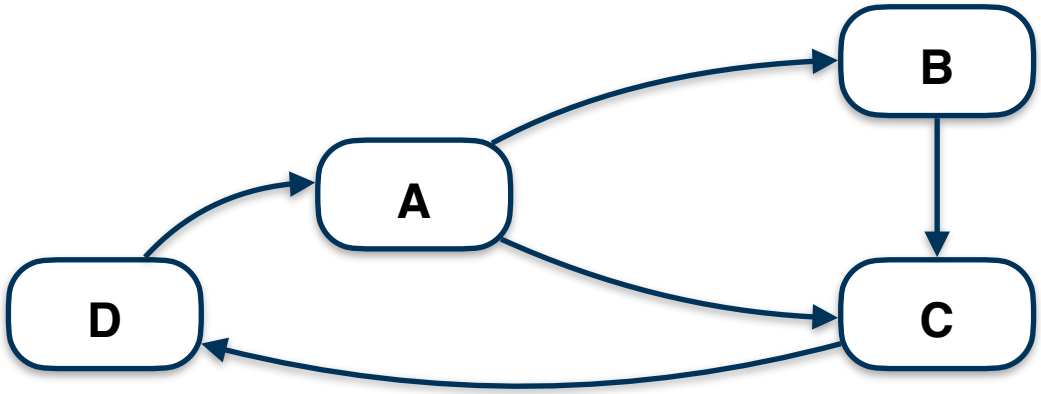
HITS Algorithmus



Berechne für den obigen Graphen die HITS-Werte nach 2 Iterationen.
Nutze 1 als Startwert für HITS.

Aufgabe 2

HITS Algorithmus



1. Iteration

	A	B	C	D
Hubs	2	1	1	1
Vorläufige Autoritäten				
Normalisierte Autoritäten				

$h_A = a_B + a_C = 1 + 1$

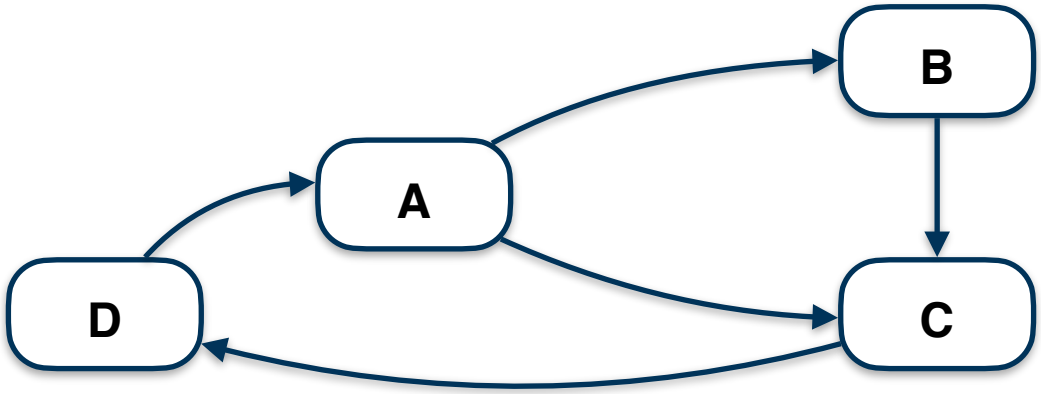
$h_B = a_C = 1$

$h_C = a_D = 1$

$h_D = a_A = 1$

Aufgabe 2

HITS Algorithmus



1. Iteration

	A	B	C	D
Hubs	2	1	1	1
Vorläufige Autoritäten	1	2	3	1
Normalisierte Autoritäten				

$a_A = h_D = 1$

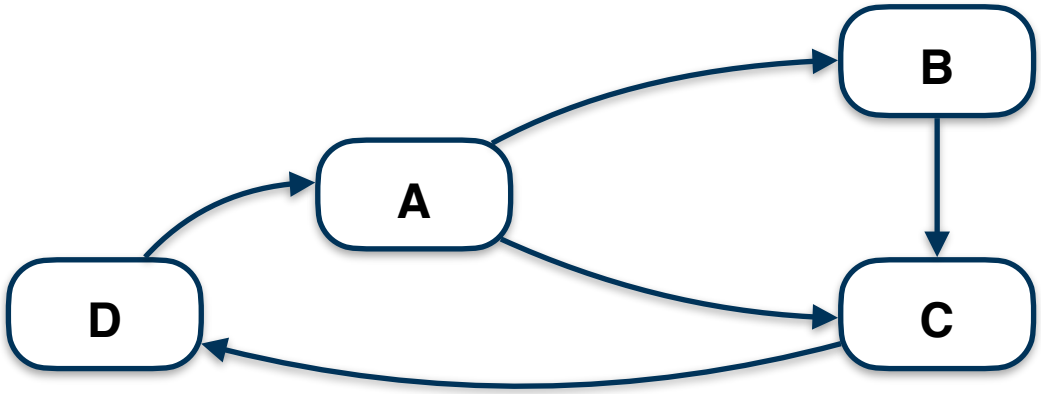
$a_B = h_A = 2$

$a_C = h_A + h_B = 2 + 1$

$a_D = h_C = 1$

Aufgabe 2

HITS Algorithmus



1. Iteration

	A	B	C	D
Hubs	2	1	1	1
Vorläufige Autoritäten	1	2	3	1
Normalisierte Autoritäten	1/3	2/3	1	1/3

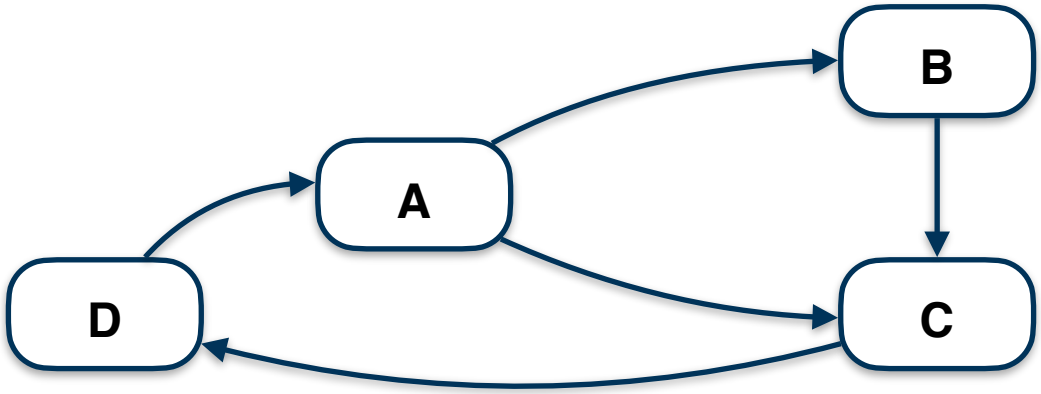
Normalisieren:

max(a) = 3

=> a_i*1/3

Aufgabe 2

HITS Algorithmus



1. Iteration

	A	B	C	D
Hubs	2	1	1	1
Vorläufige Autoritäten	1	2	3	1
Normalisierte Autoritäten	1/3	2/3	1	1/3

2. Iteration

	A	B	C	D
Hubs	5/3	1	1/3	1/3
Vorläufige Autoritäten				
Normalisierte Autoritäten				

$h_A = a_B + a_C = 2/3 + 1$

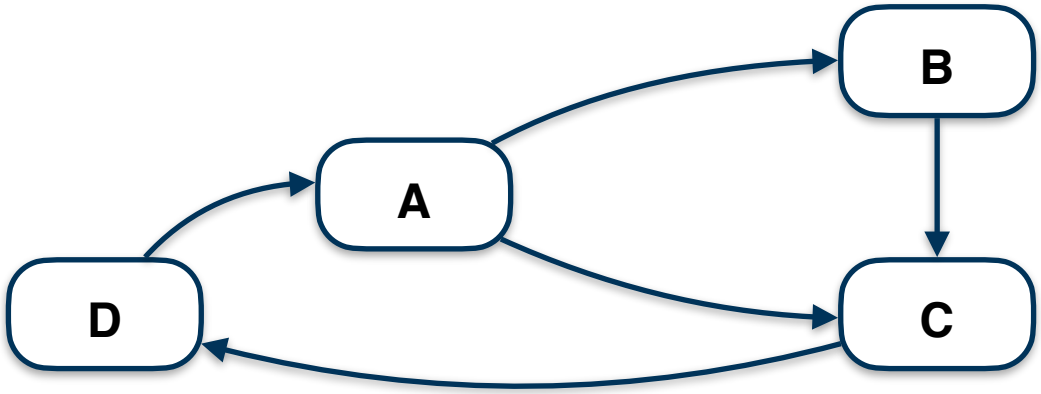
$h_B = a_C = 1$

$h_C = a_D = 1/3$

$h_D = a_A = 1/3$

Aufgabe 2

HITS Algorithmus



1. Iteration

	A	B	C	D
Hubs	2	1	1	1
Vorläufige Autoritäten	1	2	3	1
Normalisierte Autoritäten	1/3	2/3	1	1/3

2. Iteration

	A	B	C	D
Hubs	5/3	1	1/3	1/3
Vorläufige Autoritäten	1/3	5/3	8/3	1/3
Normalisierte Autoritäten				

$a_A = h_D = 1/3$

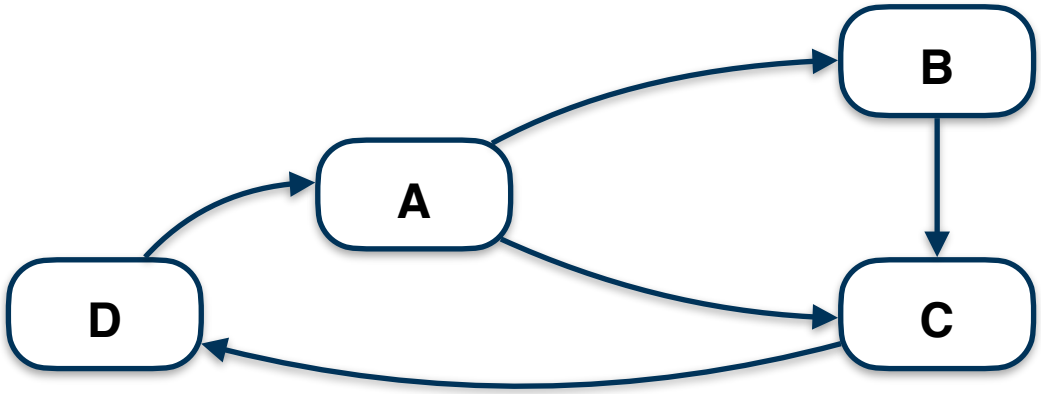
$a_B = h_A = 5/3$

$a_C = h_A + h_B = 5/3 + 1$

$a_D = h_C = 1/3$

Aufgabe 2

HITS Algorithmus



1. Iteration

	A	B	C	D
Hubs	2	1	1	1
Vorläufige Autoritäten	1	2	3	1
Normalisierte Autoritäten	1/3	2/3	1	1/3

2. Iteration

	A	B	C	D
Hubs	5/3	1	1/3	1/3
Vorläufige Autoritäten	1/3	5/3	8/3	1/3
Normalisierte Autoritäten	1/8	5/8	1	1/8

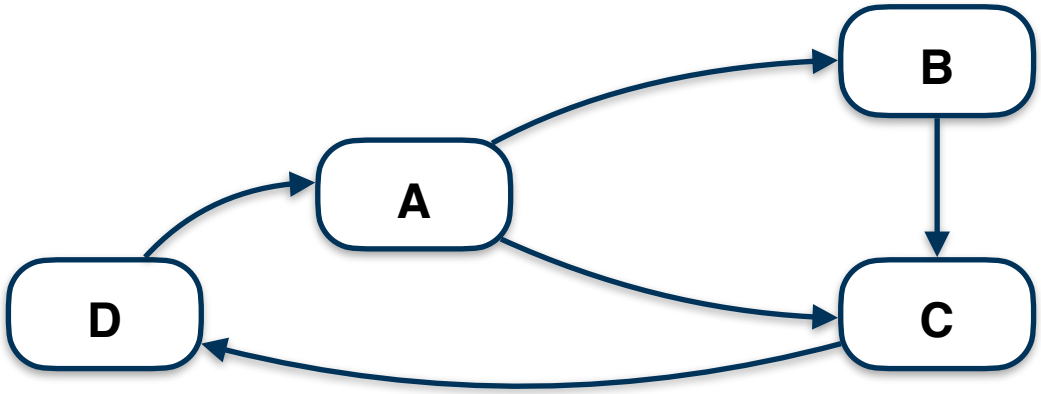
Normalisieren:

max(a) = 8/3

=> a_i*3/8

Aufgabe 2

HITS Algorithmus



1. Iteration

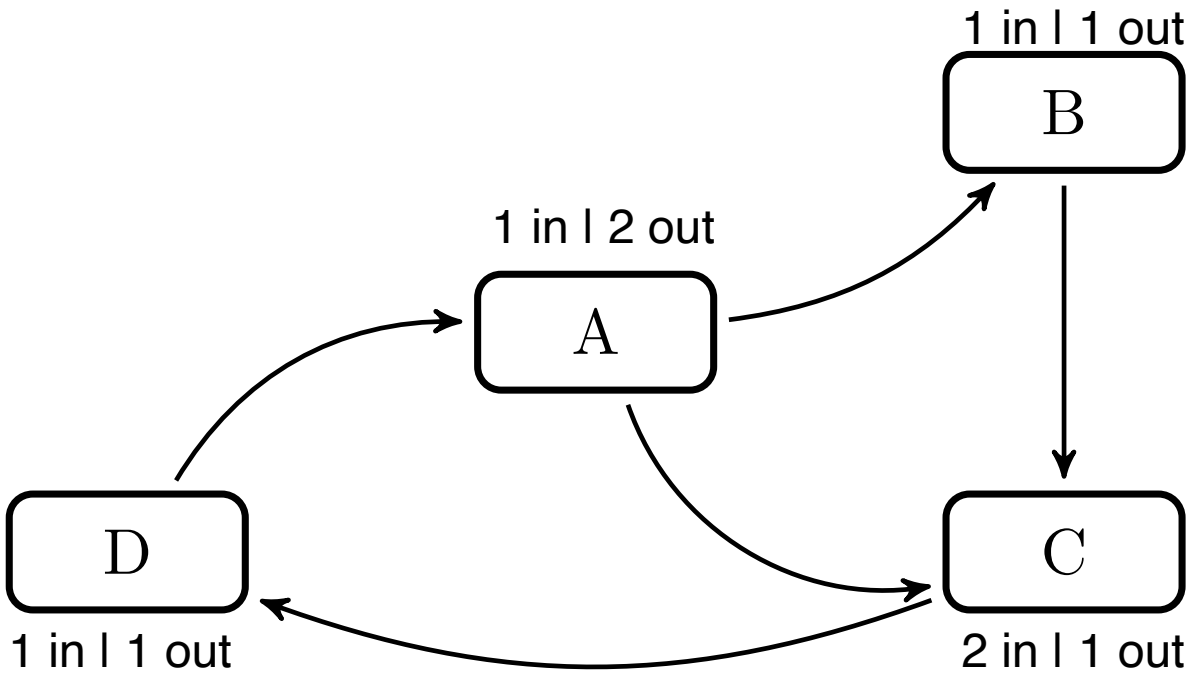
	A	B	C	D
Hubs	2	1	1	1
Vorläufige Autoritäten	1	2	3	1
Normalisierte Autoritäten	1/3	2/3	1	1/3

2. Iteration

	A	B	C	D
Hubs	5/3	1	1/3	1/3
Vorläufige Autoritäten	1/3	5/3	8/3	1/3
Normalisierte Autoritäten	1/8	5/8	1	1/8

Aufgabe 2

Pagerank - Berechnung am Graphen



Aufgabe 2
Pagerank

```
2. select VTo, 0.1/(CAST((select count(*) from pagerank)AS FLOAT))
      +0.9*sum(Beitrag)
   from(
     select e.VTo, p.Weight/
        (select count(*) from edges x where x.VFrom=e.VFrom) as Beitrag
     from edges e , pagerank p
     where e.VFrom=p.Vertex
   ) i
   group by VTo
```

Fragen?