New Media
Data Analytics and Application

Lecture 6: Data Structure of Web Crawler

Ting Wang

Outlines

- Principle of Web Crawler
- Basic Data Structure for Web Crawler
- Application of Graphs in Social Media

A brief introduction to the principle of

Web Crawler

A Review:

How to collect data from the Website of SHISU?

- Example 1 in Lecture 4

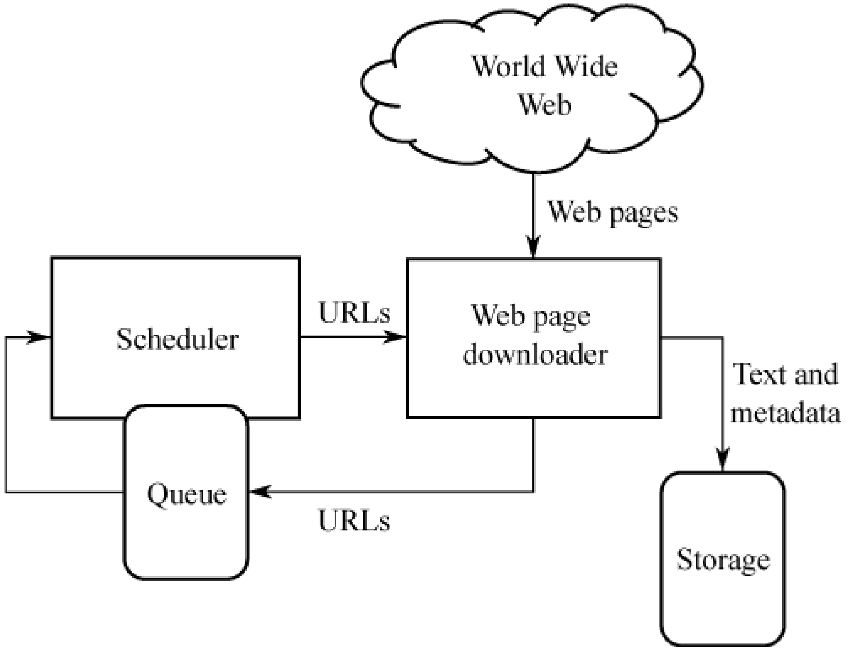
```
import urllib.request
response = urllib.request.urlopen('http://www.shisu.edu.cn/about/introducing-sisu')
HTMLText = response.read()

with open('Files/shisu.html', 'wb') as f:
    f.write(HTMLText)
```

Web Crawler 网络爬虫

an Internet robot which systematically browses the World Wide Web

Also know as:

- Web Search Engine
- Web Spider
- Web Crawling Robot



How to design a web crawler?

Based on Example 1 in Lecture 4

- Loops: While, for …
- Re-visit: If … elif…else

Web Crawler

Four Important Crawling Policies

- **selection policy** states the pages to download
- **re-visit policy** states when to check for changes to the pages
- **politeness policy** states how to avoid overloading Web sites
- **parallelization policy** states how to coordinate distributed web crawlers

Web Crawler

World Famous Web Crawler

- – Google
- – Yahoo!
- – Bing
- – Baidu
- – Soso
- – ASK

上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

the foundation of the robot for web data collection

Data Structure of Web Crawler

Data Structure

What is Data Structure

Data structure is not a data type, but a particular way of organizing data in a computer so that it can be used, stored in memory and manipulated by the program.

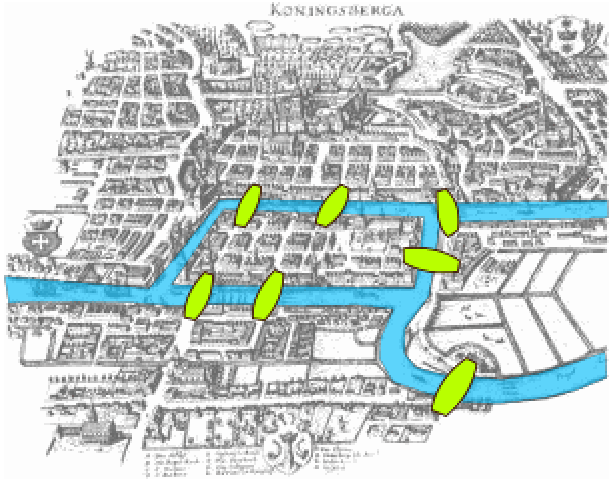
Data structure is crucial to web crawlers on data collection, storage, and analysis.

An important Data Structure for web crawlers:

GRAPH

Graph Theory: a very important branch of mathematics

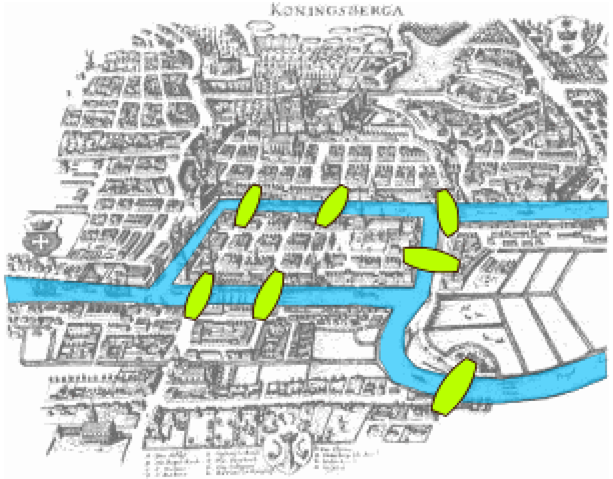
Seven Bridges of Königsberg, 1736, Euler

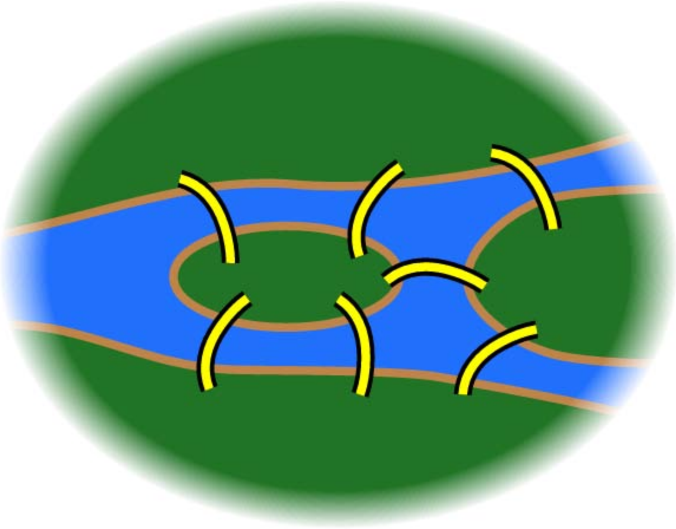


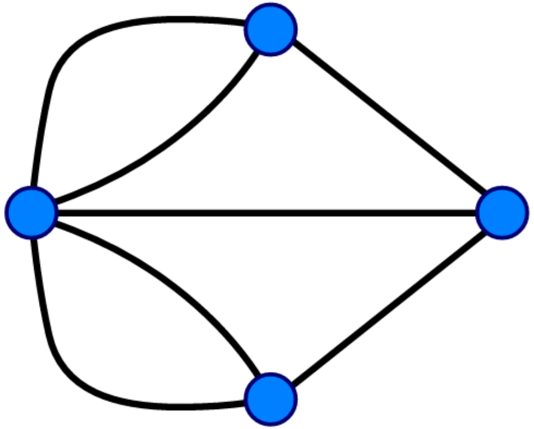
- There are 2 islands and 7 bridges that connect the islands and the mainland

- Find a path that crosses each bridge exactly once

Graph Representation of the problem







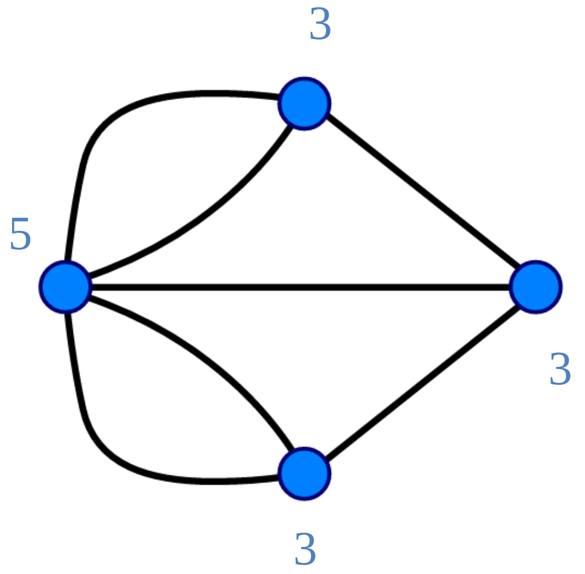
GRAPH/TREE

Data Structure

- The key to solve this problem is an ingenious graph representation

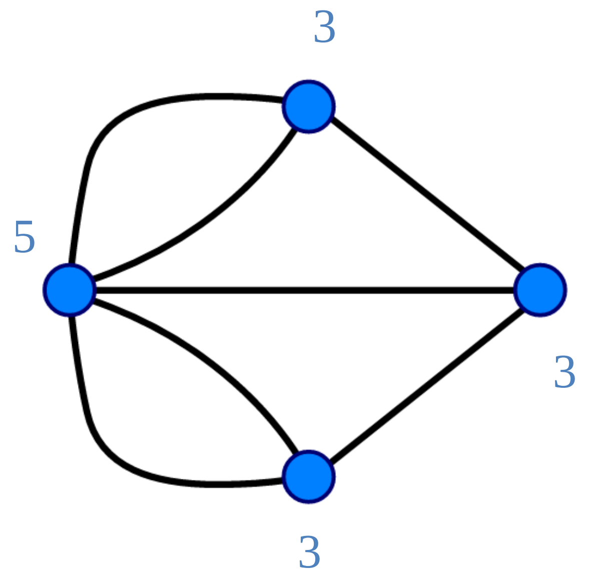
- Euler proved that since except for the starting and ending point of a walk, one has to enter and leave all other nodes, thus these nodes should have an even number of bridges connected to them

- This property does not hold in this problem



Basic Knowledge about Graph

- – Node 结点
- – Edge 边
- – Degree 度



- In modern language, Euler shows that the possibility of a walk through a graph, traversing each edge exactly once, **depends on the degrees of the nodes.**
- Euler's argument shows that a necessary condition for the walk of the desired form is that **the graph be connected and have exactly zero or two nodes of odd degree.**

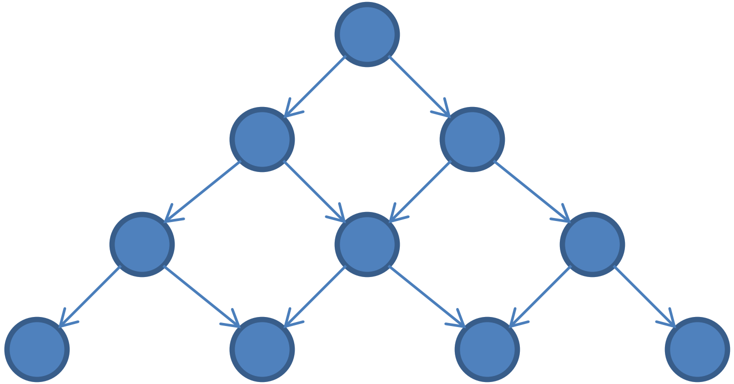
An Inference for Web Crawling deduced by the problem of Seven Bridges of Königsberg :

Re-visit is inevitable!

Question: How to set the visit path?

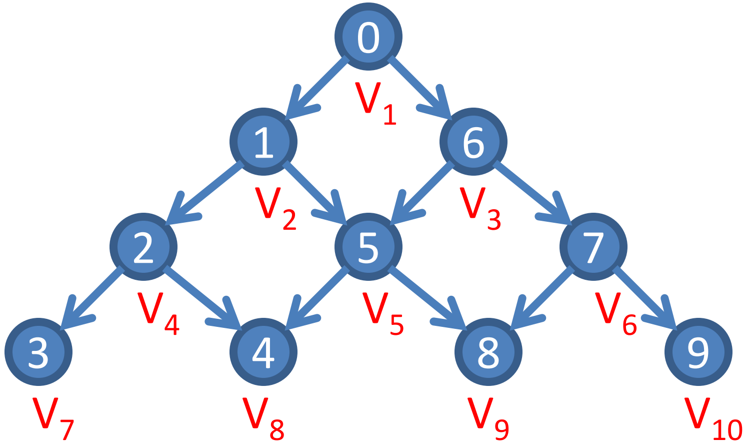
Deciding What to Search

- URL list for the websites you want to search
- Do nothing but search web pages via hyperlinks one by one



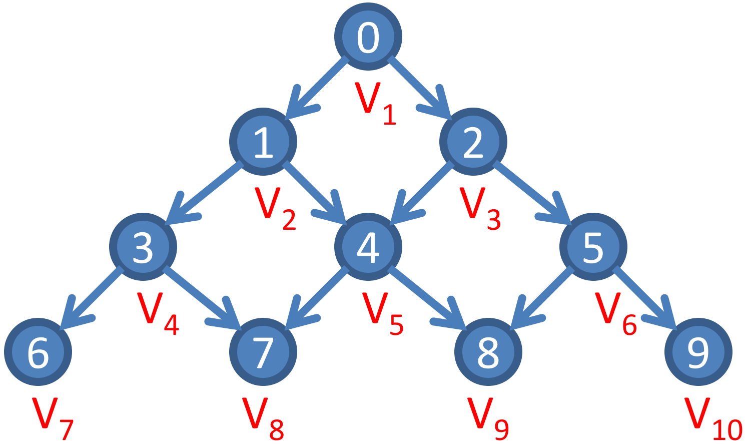
Depth-First-Search (DFS)

深度优先



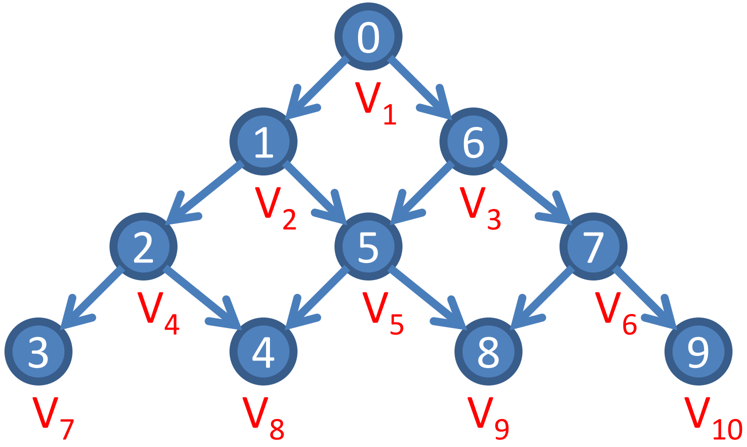
Breadth-First-Search (BFS)

广度优先



DFS Pseudo-code (1)

Recursion



Algorithm Depth-First Search (DFS): recursion

Require: Initial node v, graph/tree G(V; E)

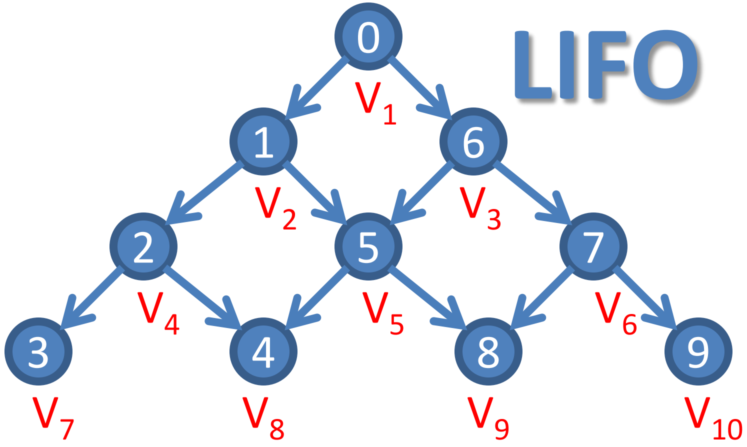
```
procedure DFS(G,vi):
    label vi as discovered
    for all edges from vi to vj in G.adjacentEdges(vi) do
        if vertex vj is not labeled as discovered then
            recursively call DFS(G,vj)
```

DFS Pseudo-code (2)

Stack

Last In First Out

LIFO



Algorithm Depth-First Search (DFS): stack

Require: Initial node v, graph/tree G(V; E), stack S

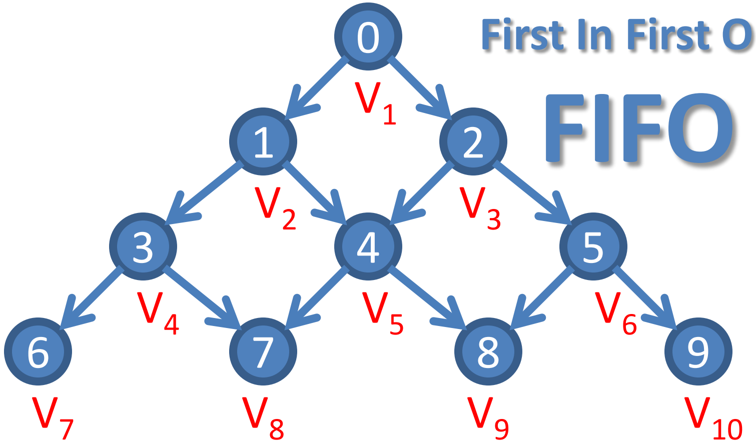
```
return An ordering on how nodes in G are visited
Push v into S;
visitOrder = 0;
while S not empty do
    node = pop from S;
    if node not visited then
        visitOrder = visitOrder +1;
        Mark node as visited with order visitOrder;
        //or print node
        Push all neighbors/children of node into S;
    end if
end while
Return all nodes with their visit order.
```

BFS Pseudo-code

Queue

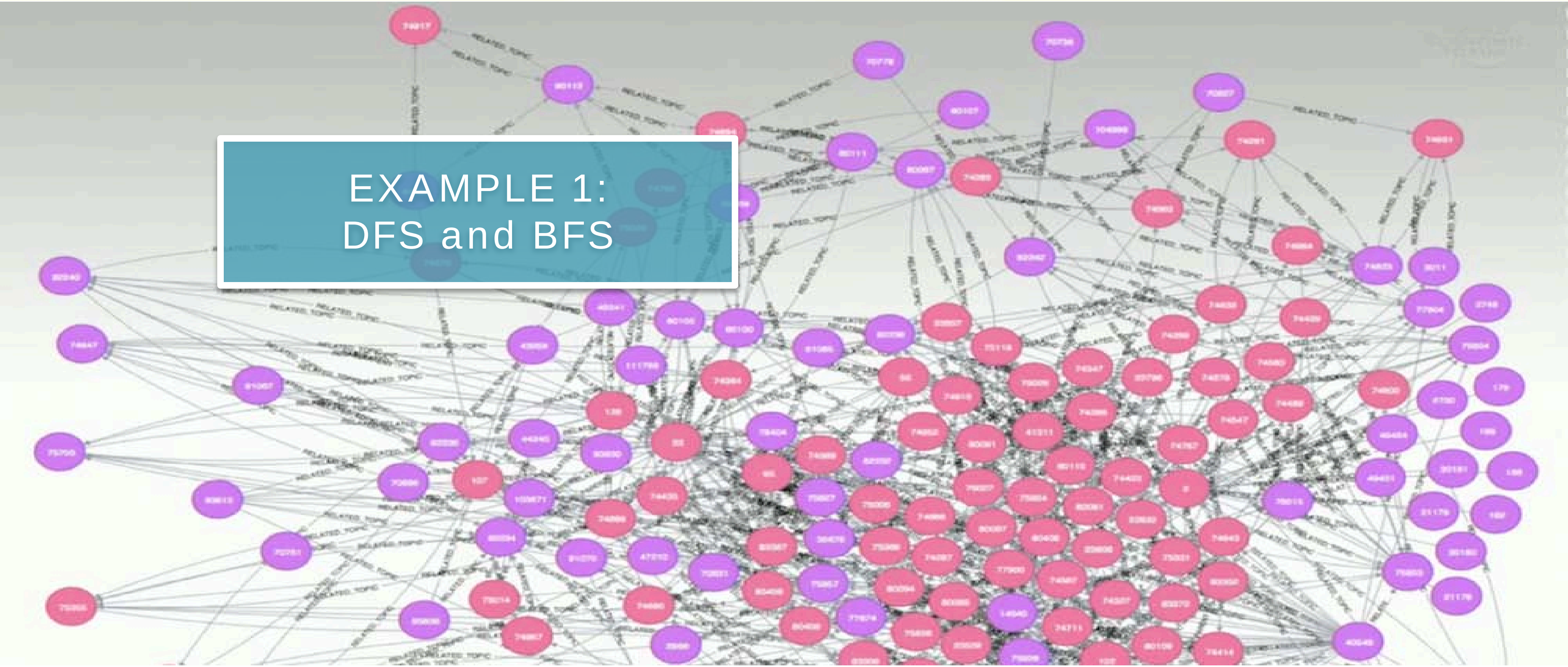
First In First Out

FIFO



```
Algorithm Breadth-First Search (BFS)
Require: Initial node v, graph/tree G(V; E), queue Q
return An ordering on how nodes are visited
Enqueue v into queue Q;
visitOrder = 0;
while Q not empty do
    node = dequeue from Q;
    if node not visited then
        visitOrder = visitOrder +1;
        Mark node as visited with order visitOrder;
        //or print node
        Enqueue all neighbors/children of node into Q;
    end if
end while
```

Data Structure

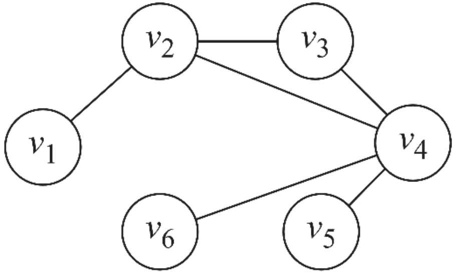


EXAMPLE 1:
DFS and BFS

Data Structure

Adjacency Matrix (a.k.a. sociomatrix) 邻接矩阵

$$A_{ij} = \begin{cases} 1\text{, if there is an edge between nodes } v_i \text{ and } v_j \\ 0\text{, otherwise} \end{cases}$$



(a) Graph

	v1	v2	v3	v4	v5	v6
v1	0	1	0	0	0	0
v2	1	0	1	1	0	0
v3	0	1	0	1	0	0
v4	0	1	1	0	1	1
v5	0	0	0	1	0	0
v6	0	0	0	1	0	0

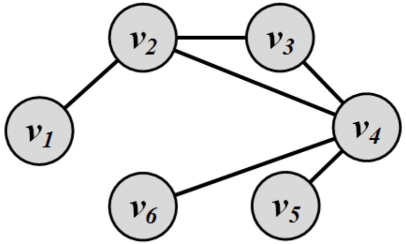
(b) Adjacency Matrix

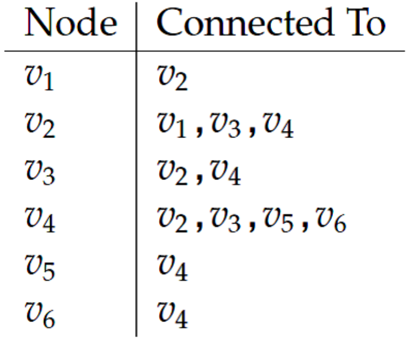
Diagonal Entries are self-links or loops

Social media networks have very sparse Adjacency matrices

Adjacency List 邻接表

- In an adjacency list for every node, we maintain a list of all the nodes that it is connected to
- The list is usually sorted based on the node order or other preferences

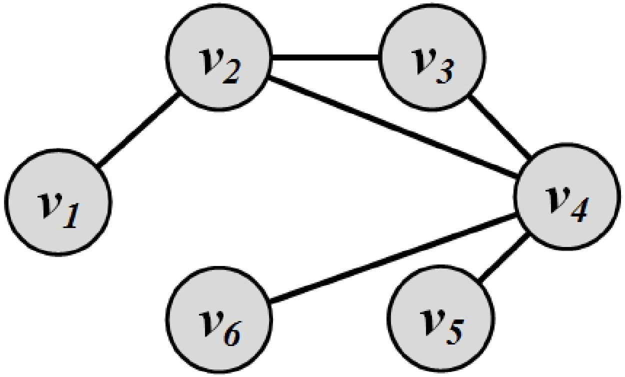




Node	Connected To
v_1	v_2
v_2	v_1, v_3, v_4
v_3	v_2, v_4
v_4	v_2, v_3, v_5, v_6
v_5	v_4
v_6	v_4

Edge List 边列表

- In this representation, each element is an edge and is represented as (v_i; v_j), denoting that node v_i is connected to node v_j.
- Since social media networks are sparse, both the adjacency list and edge list representations save significant space.



(v_1,v_2)

(v_2,v_3)

(v_2,v_4)

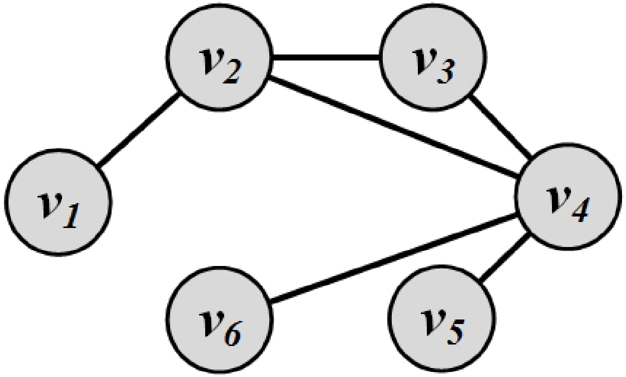
(v_3,v_4)

(v_4,v_5)

(v_4,v_6)

Save in the Data Base

- Using Edge List



V_ID	V_Name
1	V1
2	V2
3	V3
4	V4
5	V5
6	V6
…	…

E_ID	Vi_ID	Vj_ID
1	1	2
2	2	3
3	2	4
4	3	4
5	4	5
6	4	6
…	…	…

Save Web URL in the Data Base for Crawlers

- Using Edge List

URL_ID	URL
1	www.sina.com
2	www.weibo.com
3	www.weibo.com/tv
4	d.weibo.com/?topnav=1&mod=logo
5	weibo.com/u/3941468498?refer_flag=1028035010_&is_all=1
6	http://weibo.com/u/1766565543?refer_flag=1028035010_&is_all=1
…	…

E_ID	Vi_ID	Vj_ID
1	1	2
2	2	3
3	2	4
4	3	4
5	4	5
6	4	6
…	…	…

Types of Graphs *图的类型*

1. NULL Graph (no nodes, so no edge)

$$G(V,E), \quad V = E = \emptyset.$$

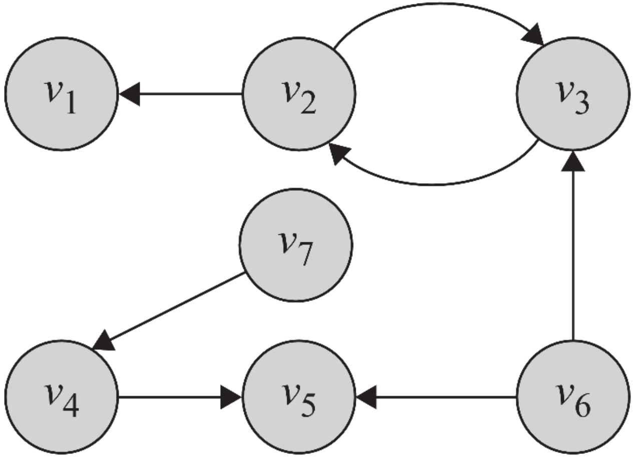
2. Empty Graph (no edge, but maybe has nodes)

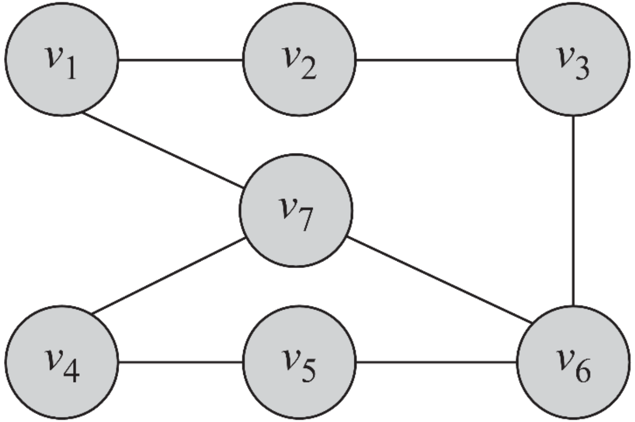
$$G(V,E), \quad E = \emptyset.$$

3. Directed/Undirected/Mixed Graphs

有向图/无向图/混合图

Web sites are directed graphs

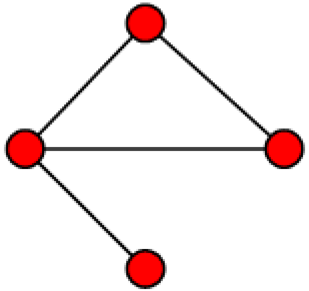




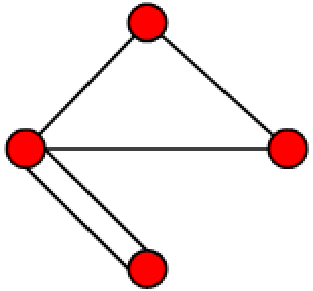
4. Simple Graph / Multigraph

简单图 / 多重图

Many web sites are Multigraphs



Simple graph



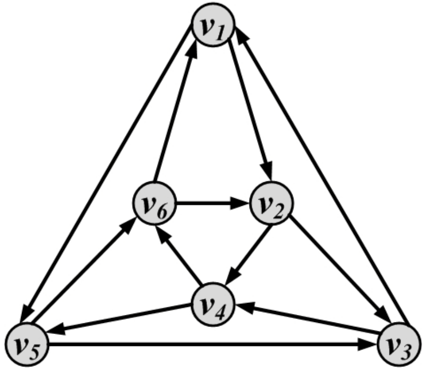
Multigraph

Connectivity in Graphs *图的连通性*

- Adjacent nodes and Incident Edges 相邻节点和相邻边
  - –Two nodes are adjacent if they are connected via an edge.
  - –Two edges are incident, if they share on end-point

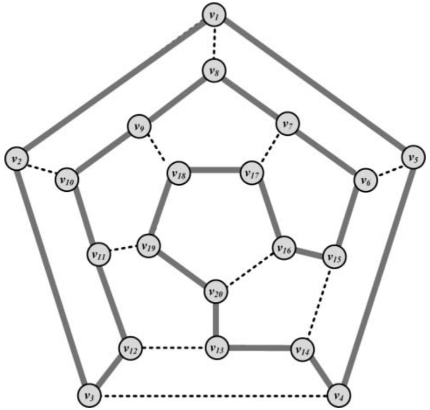
Eulerian Tour 欧拉环路

- All edges are traversed only once
  - – Konigsberg bridges

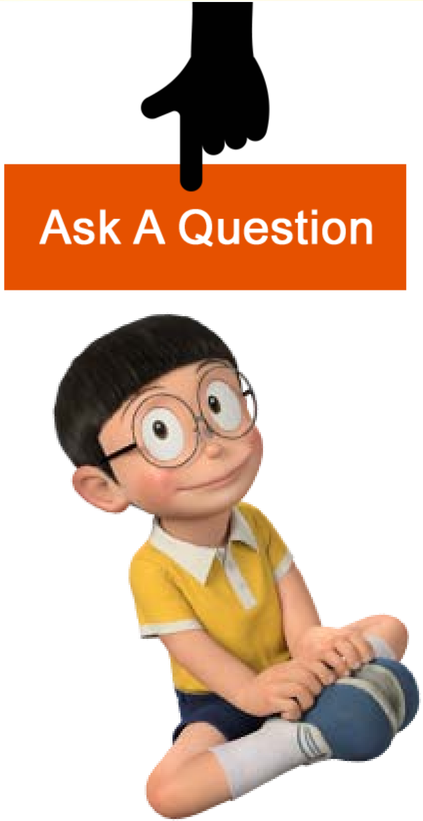


Hamiltonian Cycle 汉密尔顿回路

- A cycle that visits all nodes



Data Structure

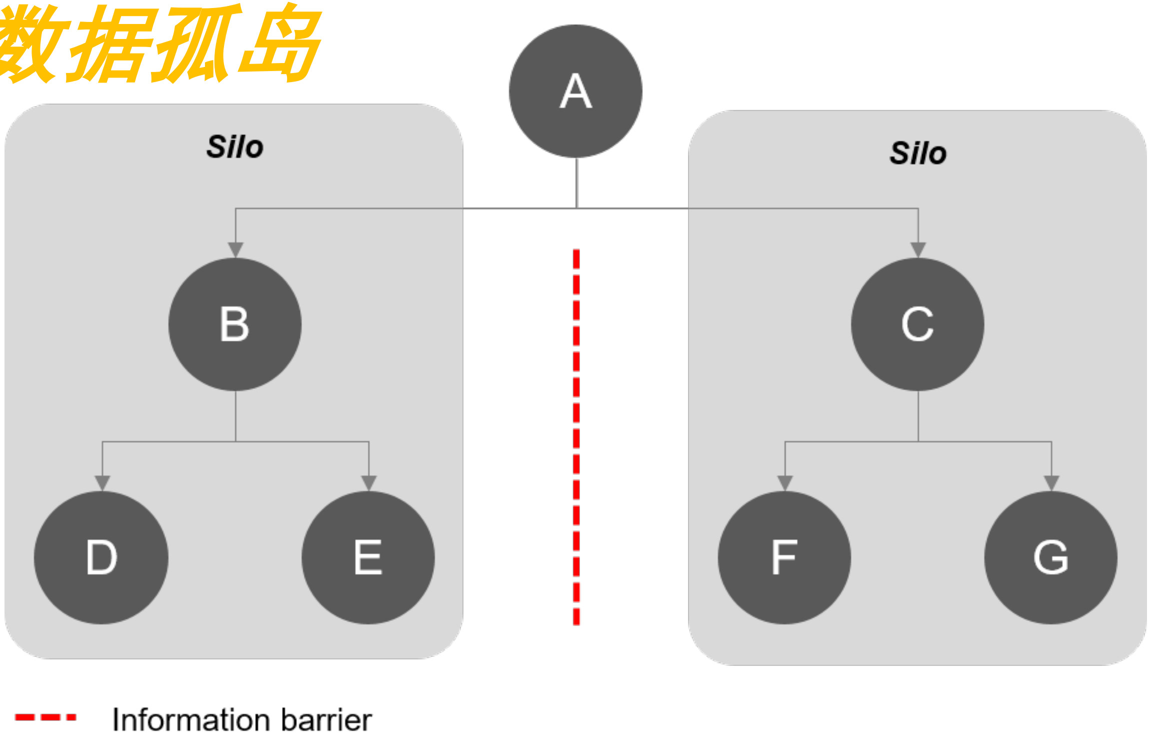


How to avoid endless loops in web crawling?

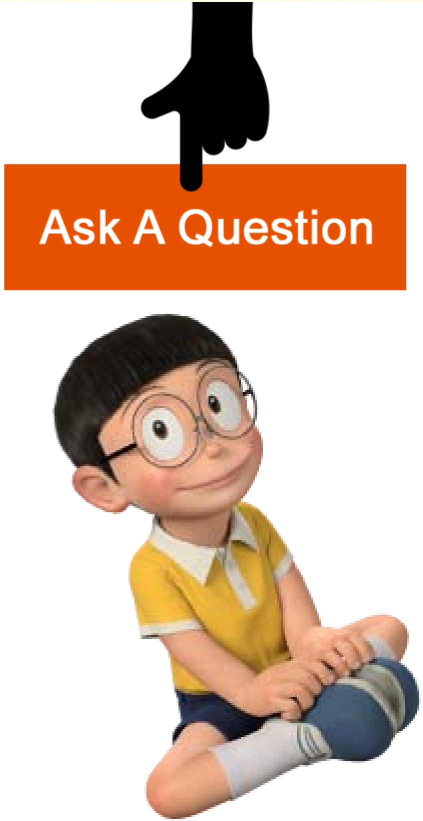
- *–Reject short-time re-visit*
- *–Accept long-time re-visit*

Graph Connectivity and Data Silos

图的连通性与数据孤岛



Data Structure



How to jump into Data Silos?

- *–To set more different start web URLs in initialization.*
- *–Multi-thread Process*

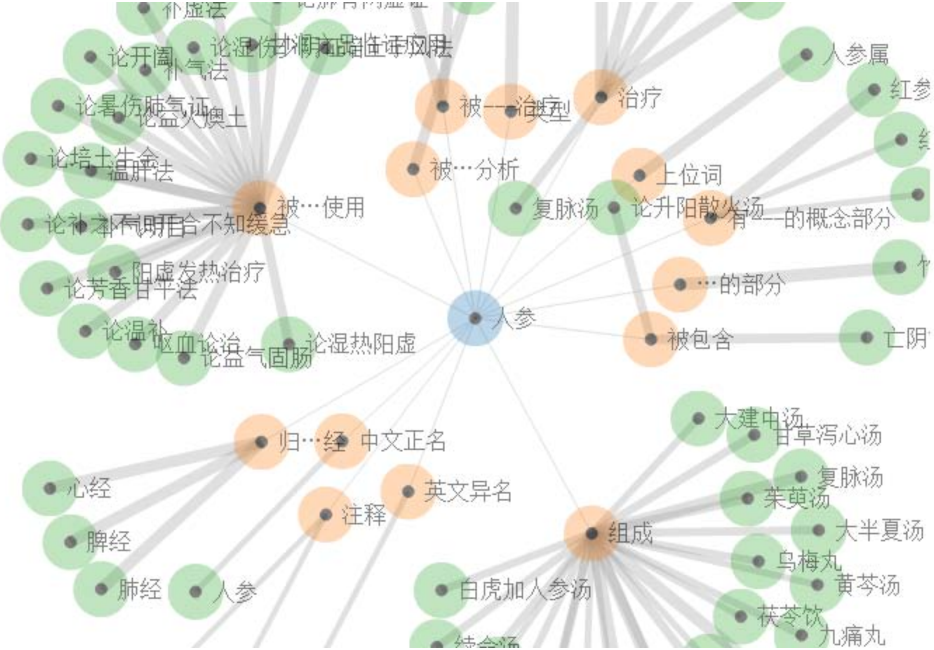
application of graphs in relationships between people, news, and others

Application of Graphs in Social Media

Application of Graphs in Social Media

Semantic Networks / Knowledge Graph

语义网络 / 知识图谱



Application of Graphs in Social Media

- Social Media like Linkedin, research gate

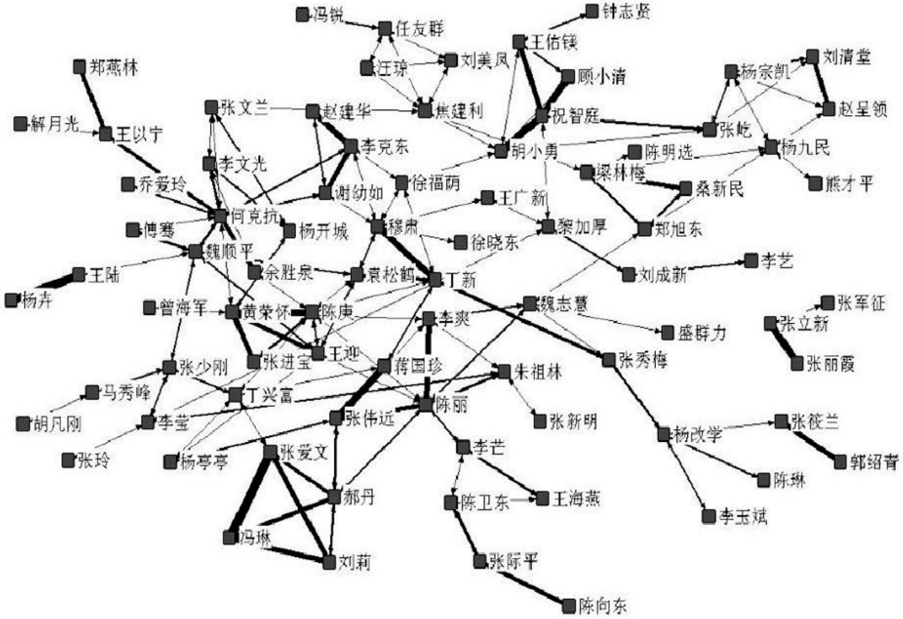


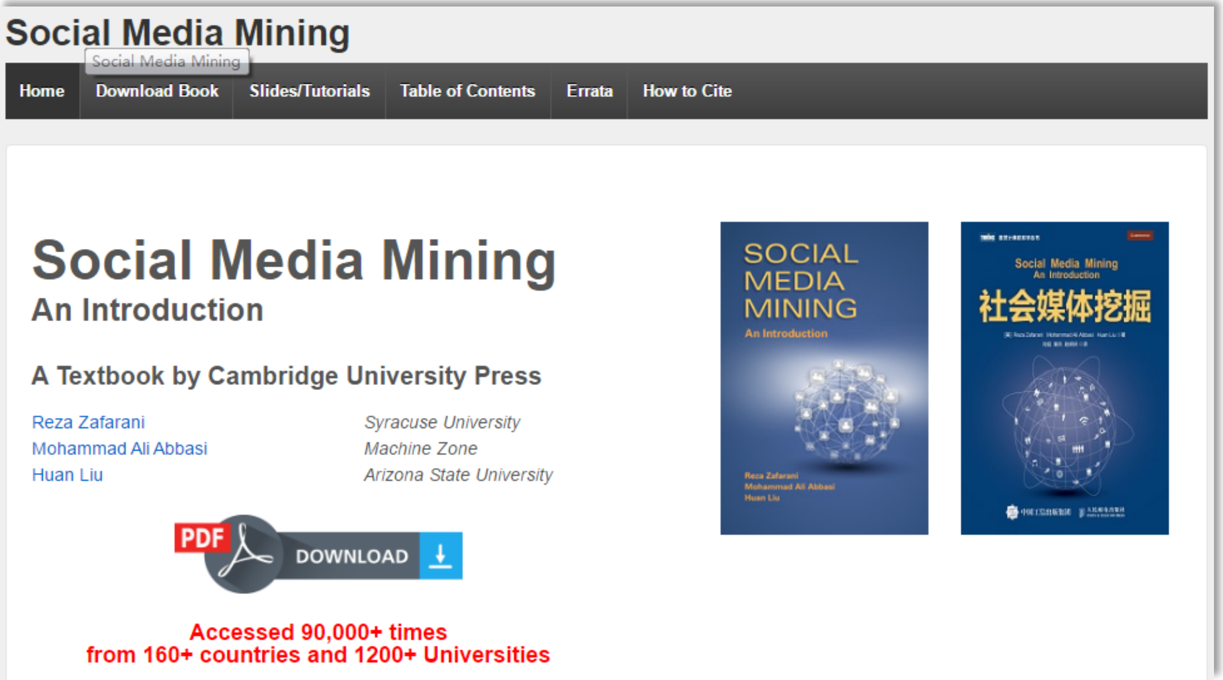
图 1　我国教育技术学者合著网络知识图谱

Reference

Reference

Social Media Mining

– http://dmml.asu.edu/smm/



Social Media Mining

Home | Download Book | Slides/Tutorials | Table of Contents | Errata | How to Cite

Social Media Mining

An Introduction

A Textbook by Cambridge University Press

Reza Zafarani	*Syracuse University*
Mohammad Ali Abbasi	*Machine Zone*
Huan Liu	*Arizona State University*

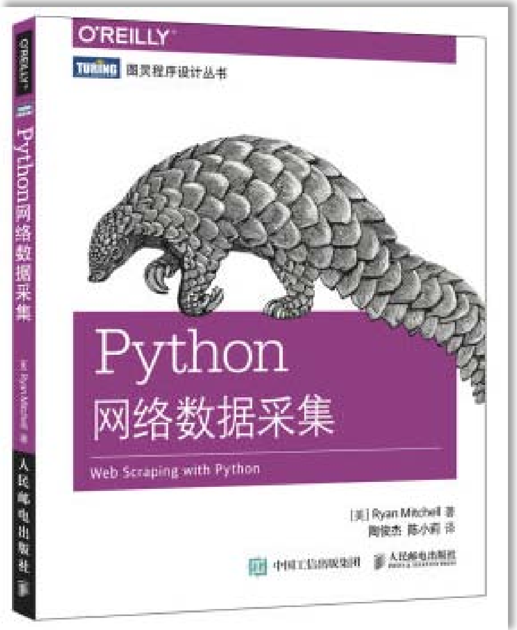
PDF DOWNLOAD

Accessed 90,000+ times
from 160+ countries and 1200+ Universities

Reference

Python网络数据采集

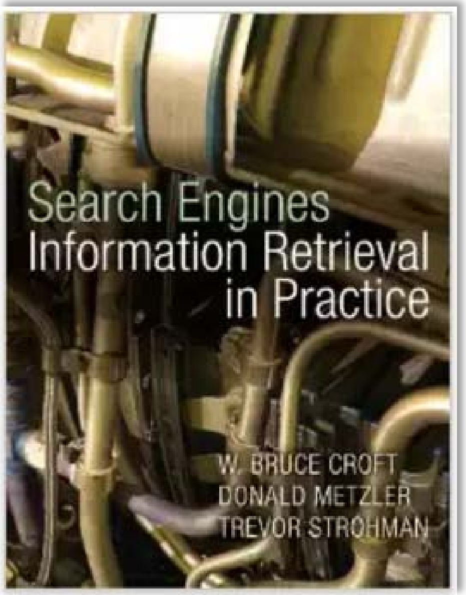
- https://item.jd.com/11896401.html
- http://download.csdn.net/detail/u010309742/9647121?web=web



Reference

search engine information retrieval in practice

- http://www.search-engines-book.com/
- http://www.amazon.com/Search-Engines-Information-Retrieval-Practice/dp/0136072240


Search Engines
Information Retrieval
in Practice
W. BRUCE CROFT
DONALD METZLER
TREVOR STROHMAN

Homework

Homework

A Report for Designing Your Web Crawler

- Should contain:

1. Deciding What to Search
    - A website list you want to search, and why

2. Approaches for Information Acquisition
    - a selection between DFS or BFS, or both, and why

Homework

3. Data Base
    - A form about database designing
    - ER-diagrams for database of web crawler
    - SQL script for table creation, drop and initialization

4. Software Engineering Documents
    - UML diagrams for the software of web crawler

    (Use Case, Activity, Sequence, and so on)

Homework

- Important Dates:
  - – Submission Deadline: before November 8th, 2016
  - – Presentation: November 9th, 2016
  - – Coding/Testing Deadline: November 29th, 2016
  - – Demonstration: November 30th, 2016

The End of Lecture 6

Thank You

http://www.wangting.ac.cn