CS483-09 Elementary Graph Algorithms

Instructor: Fei Li

Room 443 ST II

Office hours: Tue. & Thur. 1:30pm - 2:30pm or by appointments

lifei@cs.gmu.edu with subject: CS483

http://www.cs.gmu.edu/∼ lifei/teaching/cs483_fall07/

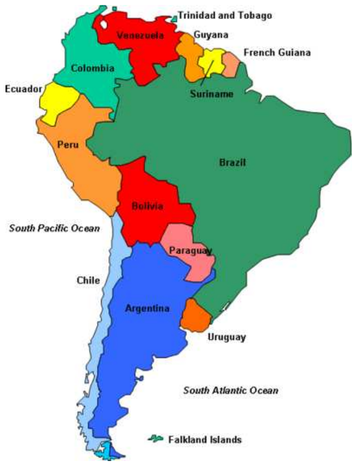
Based on “Introduction to Algorithms” by T. Cormen, C. Leiserson, R. Rivest, and C. Stein and

“Algorithms” by S. Dasgupta, C. Papadimitriou, and U. Vazirani.

Outline

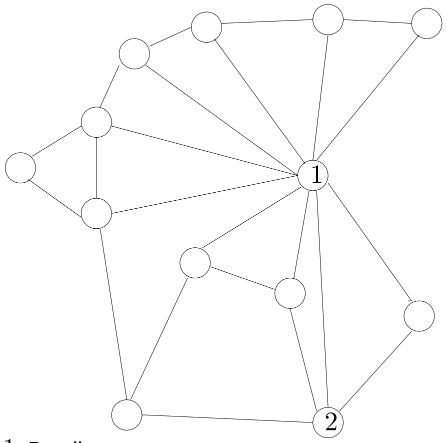
- Representation of Graphs
- Breath-first Search
- Depth-first Search
- Topological Sort

Why Graphs?



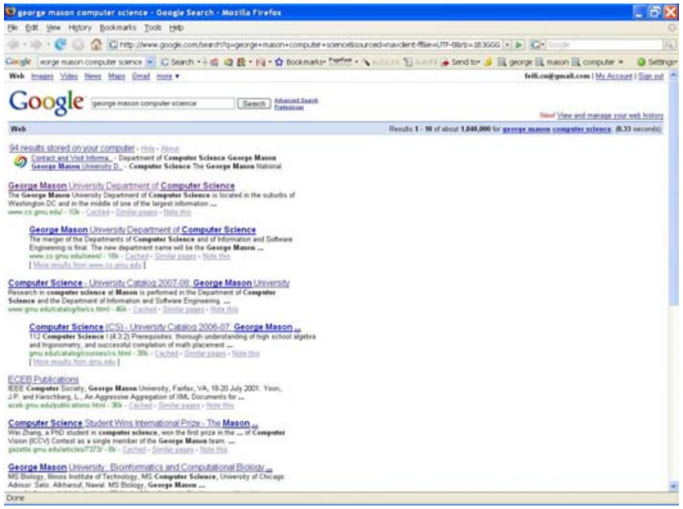
http://www.transitionsabroad.com/images/maps/south_america_map.gif

Why Graphs?



1 Brazil
2 Argentina

Why Graphs?



george mason computer science - Google Search - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.google.com/search?q=george+mason+computer+science&sourceid=navclient-ff&ie=UTF-8&rlz=1B3GGG

Web Images Video News Maps Gmail more ▾ — felli.cs@gmail.com | My Account | Sign out

Google [george mason computer science] Search — Advanced Search, Preferences

New! View and manage your web history

Web — Results 1 - 10 of about 1,840,000 for george mason computer science. (0.33 seconds)

94 results stored on your computer - Hide - About
Contact and Visit Informa... - Department of Computer Science George Mason
George Mason University D... - Computer Science The George Mason National

George Mason University Department of Computer Science
The George Mason University Department of Computer Science is located in the suburbs of Washington DC and in the middle of one of the largest information ...
www.cs.gmu.edu/ - 10k - Cached - Similar pages - Note this

George Mason University Department of Computer Science
The merger of the Departments of Computer Science and of Information and Software Engineering is final. The new department name will be the George Mason ...
www.cs.gmu.edu/news/ - 18k - Cached - Similar pages - Note this
[More results from www.cs.gmu.edu]

Computer Science - University Catalog 2007-08: George Mason University
Research in computer science at Mason is performed in the Department of Computer Science and the Department of Information and Software Engineering. ...
www.gmu.edu/catalog/tw/cs.html - 46k - Cached - Similar pages - Note this

Computer Science (CS) - University Catalog 2006-07: George Mason ...
112 Computer Science I (4:3:2) Prerequisites: thorough understanding of high school algebra and trigonometry, and successful completion of math placement ...
gmu.edu/catalog/courses/cs.html - 38k - Cached - Similar pages - Note this
[More results from gmu.edu]

ECEB Publications
IEEE Computer Society, George Mason University, Fairfax, VA, 18-20 July 2001. Yoon, J.P. and Kerschberg, L., An Aggressive Aggregation of XML Documents for ...
eceb.gmu.edu/publications.html - 36k - Cached - Similar pages - Note this

Computer Science Student Wins International Prize - The Mason ...
Wei Zhang, a PhD student in computer science, won the first prize in the ... of Computer Vision (ICCV) Contest as a single member of the George Mason team. ...
gazette.gmu.edu/articles/7373/ - 8k - Cached - Similar pages - Note this

George Mason University : Bioinformatics and Computational Biology ...
MS Biology, Illinois Institute of Technology, MS Computer Science, University of Chicago Advisor: Saleh, Alkharouf, Nawal. MS Biology, George Mason ...

Done

Why Graphs?

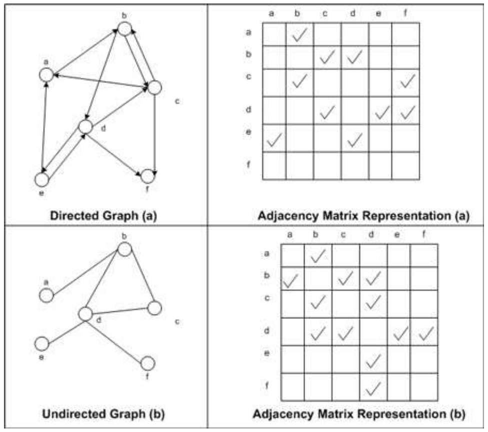
http://adcentered.typepad.com/photos

Graphs

- A graph $G = (V, E)$ is specified by a set of vertices (nodes) V and edges E between selected pairs of vertices.
- Edges are symmetric — *undirected graph*
- Directions over edges — *directed graph*
- Examples: political maps, exam conflicts, World Wide Web, etc.

Graph Representation

- Adjacency-matrix



Directed Graph (a)

	a	b	c	d	e	f
a		✓				
b			✓	✓		
c		✓				✓
d			✓		✓	✓
e	✓			✓		
f						

Adjacency Matrix Representation (a)

Undirected Graph (b)

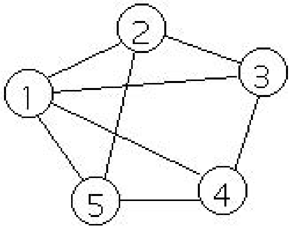
	a	b	c	d	e	f
a		✓				
b	✓		✓	✓		
c		✓		✓		
d		✓	✓		✓	✓
e				✓		
f				✓		

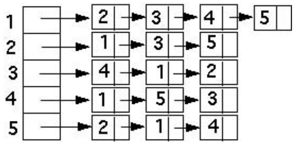
Adjacency Matrix Representation (b)

http://msdn2.microsoft.com/en-us/library/

Graph Representation

- Adjacency-list





1 → 2 → 3 → 4 → 5
2 → 1 → 3 → 5
3 → 4 → 1 → 2
4 → 1 → 5 → 3
5 → 2 → 1 → 4

Graph Traversal is Important

Exploring a graph is rather like navigating a maze.

Which parts of the graph are reachable from a given vetex?

`http://www.sheffordtown.co.uk/maze/index.html`

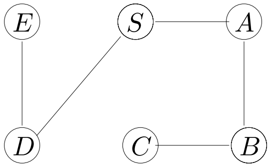
Outline

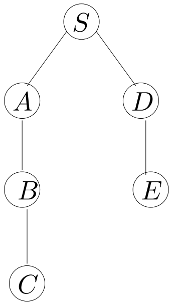
- Representation of Graphs
- Breath-first Search
- Depth-first Search
- Topological Sort

Breath-first Search (BFS)

BFS

1. Identifies all the vertices of a graph that can be reached from a designated starting point, and
2. Finds explict paths via a depth-first search tree.





Breath-first Search (BFS)

Input: Graph $G = (V, E)$, directed or undirected; vertex $s \in V$

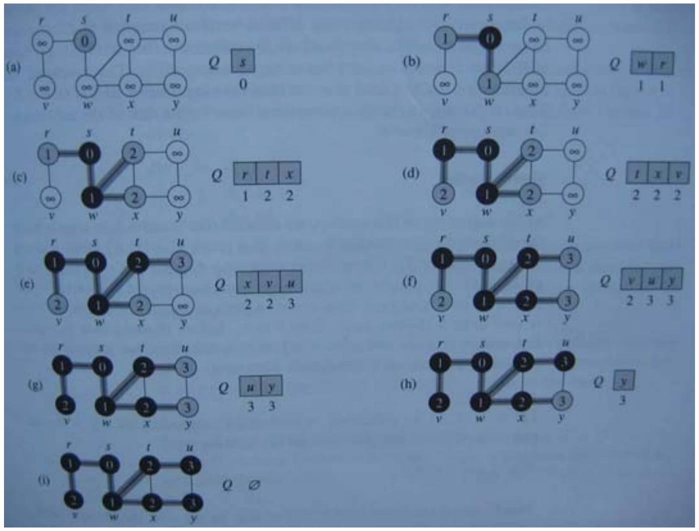
Output: For all vertices u reachable from s, $d(u)$ is set to the distance from s to u

Intuition: Proceed layer by layer

Algorithm 0.1: BFS(G, s)

```
for $\forall u \in V$
  $d(u) = \infty$
$d(s) = 0$

$Q = [s]$
while $Q \neq \emptyset$
  $u =$ Pop $(Q)$
  for $(u, v) \in E$
    if $d(v) = \infty$
      then Push $(Q, v)$
           $d(v) = d(u) + 1$
```



Breath-first Search (BFS)

- The correctness proof: Use an induction method
- The overall running time of BFS is $O(|V| + |E|)$.
  - Each vertex is put on the queue exactly once, when it is first encountered, so there are $2 \cdot |V|$ queue operations.
  - Over the course of execution, this loop looks at each edge once (in directed graphs) or twice (in undirected graphs), and therefore takes $O(|E|)$ time.

Outline

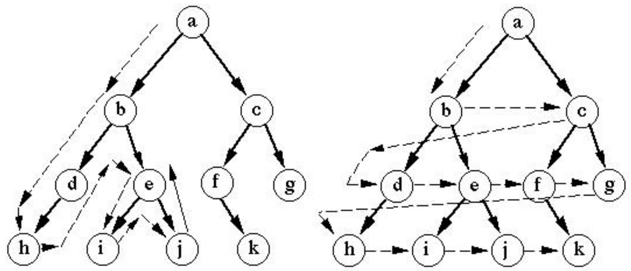
- Representation of Graphs
- Breath-first Search
- Depth-first Search
- Topological Sort

Depth-first Search

Input: Graph $G = (V, E)$, directed or undirected; vertex $s \in V$

Output: All vertices u reachable from s in timestamps of visiting

Intuition: Explore each vertex as much as you can



Depth-first search

Breadth-first search

http://www.cse.unsw.edu.au/ billw/Justsearch1.gif

Depth-first Search (DFS)

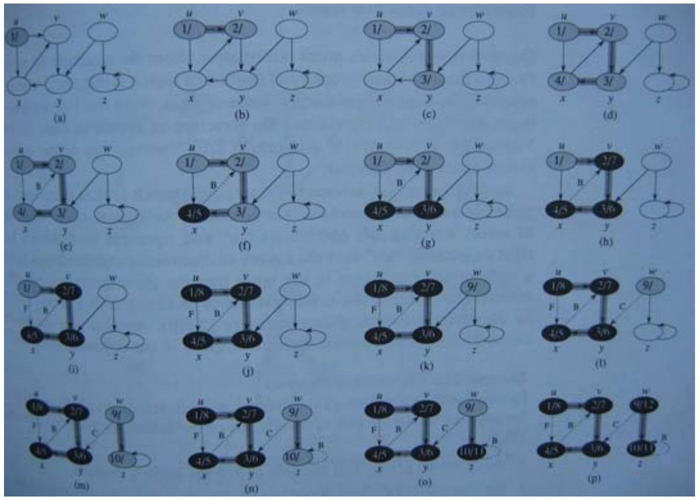
$\pi[u]$: the parent of a node u.

$time[u]$: timestamp when u is first discovered.

```
Algorithm 0.2: DFS($G(V, E)$)
 for each vertex $u \in V(G)$
   do color$[u] \leftarrow$ WHITE
        $\pi[u] \leftarrow$ NIL
 time $\leftarrow 0$
 for each vertex $u \in V(G)$
   do if color$[u] =$ WHITE
   then DFS-VISIT(u)
```

Algorithm 0.3: DFS-VISIT(u)

```
color$[u] \longleftarrow$ GRAY
$//$White vertex u has just been discovered.
$d[u] \longleftarrow$ time $\longleftarrow$ time $+ 1$
for each $v \in$ Adj$[u]$
$//$Explore edge $(u, v)$.
  do if color$[v] =$ WHITE
  then $\pi(u) \longleftarrow u$
  DFS-VISIT(v)
color$[u] \longleftarrow$ BLACK
$//$Blacken u; it is finished.
$d[u] \longleftarrow$ time $\longleftarrow$ time $+ 1$
```


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)
(k)
(l)
(m)
(n)
(o)
(p)
u
v
w
x
y
z
1/8
2/7
4/5
3/6
9/12
10/11
B
F
C