Objectives

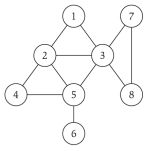
- Graphs
- Graph Connectivity, Traversal
- BFS & DFS Implementations, Analysis

Review

- What is a heap?
- When is it useful?

Undirected Graphs G = (V, E)

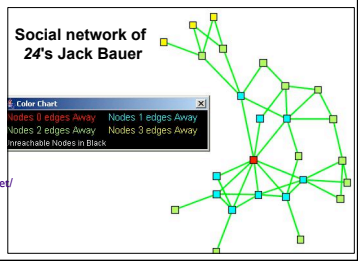
- V = nodes (vertices)
- E = edges between pairs of nodes
- Captures pairwise relationship between objects
- Graph size parameters: n = |V|, m = |E|



V = { 1, 2, 3, 4, 5, 6, 7, 8 }
E = { 1-2, 1-3, 2-3, 2-4, 2-5, 3-5, 3-7, 3-8, 4-5, 5-6 }
n = 8
m = 11

Social Networks

- Node: people; Edge: relationship between 2 people
- *Everything Bad Is Good for You: How Today's Popular Culture Is Actually Making Us Smarter*
  - Television shows have complex plots, complex social networks



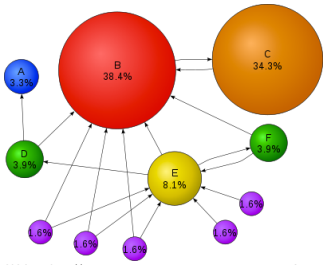
http://www.cs.duke.edu/csed/harambeenet/modules.html

Facebook: Visualizing Friends

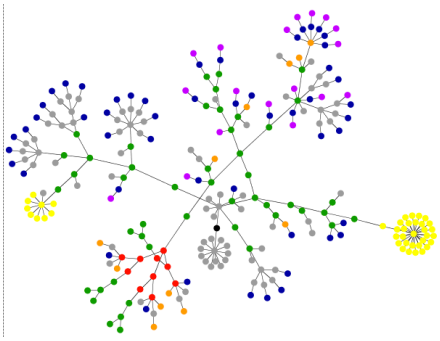
http://www.facebook.com/notes/facebook-engineering/visualizing-friendships/469716398919

World Wide Web

- Node: web page
- Edge: hyperlink from one page to another
- Uses
  - Page rank (shown)
  - Usability



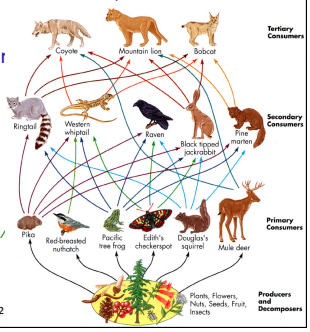
Graph of Web Page www.wlu.edu



Ecological Food Web

- Food web graph
  - Node = species
  - Edge = from prey to pr

Directed Graph:



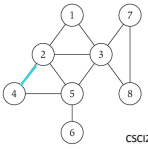
Reference:
https://www.msu.edu/course/isb/202/ebertmay/images/foodweb.jpg

Example Graph Applications

Graph	Nodes	Edges
transportation	street intersections	highways
communication	computers	fiber optic cables
World Wide Web	web pages	hyperlinks
social	people	relationships
food web	species	predator-prey
software systems	functions	function calls
scheduling	tasks	precedence constraints
circuits	gates	wires

Graph Representation: Adjacency Matrix

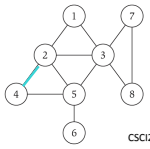
- n×n matrix with A_{uv} = 1 if (u, v) is an edge
  - Two representations of each edge (symmetric matrix)
  - Space?
  - Checking if (u, v) is an edge?
  - Identifying all edges?



	1	2	3	4	5	6	7	8
1	0	1	1	0	0	0	0	0
2	1	0	1	1	1	0	0	0
3	1	1	0	0	1	0	1	1
4	0	1	0	0	1	0	0	0
5	0	1	1	1	0	1	0	0
6	0	0	0	0	1	0	0	0
7	0	0	1	0	0	0	0	1
8	0	0	1	0	0	0	1	0

Graph Representation: Adjacency Matrix

- n×n matrix with A_{uv} = 1 if (u, v) is an edge
  - Two representations of each edge (symmetric matrix)
  - Space: $\Theta(n^2)$
  - Checking if (u, v) is an edge: $\Theta(1)$ time
  - Identifying all edges: $\Theta(n^2)$ time

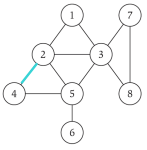


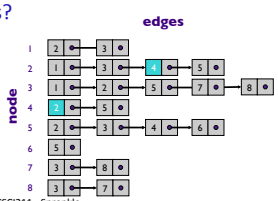
	1	2	3	4	5	6	7	8
1	0	1	1	0	0	0	0	0
2	1	0	1	1	1	0	0	0
3	1	1	0	0	1	0	1	1
4	0	1	0	0	1	0	0	0
5	0	1	1	1	0	1	0	0
6	0	0	0	0	1	0	0	0
7	0	0	1	0	0	0	0	1
8	0	0	1	0	0	0	1	0

Graph Representation: Adjacency List

- Node indexed array of lists
  - Two representations of each edge
  - Space?
  - Checking if (u, v) is an edge?
  - Identifying all edges?

What are the extremes?

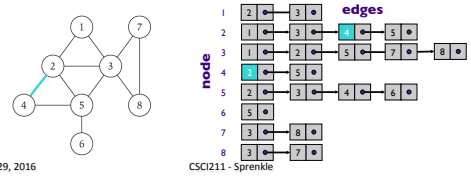




Graph Representation: Adjacency List

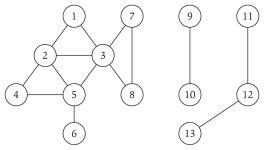
- Node indexed array of lists
  - Two representations of each edge
  - Space = 2m + n = O(m + n)
  - Checking if (u, v) is an edge takes O(deg(u)) time
  - Identifying all edges takes Θ(m + n) time

degree = number of neighbors of u



Paths and Connectivity

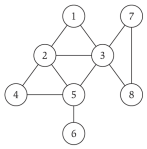
- Def. A ***path*** in an undirected graph G = (V, E) is a sequence P of nodes v_1, v_2, …, v_{k-1}, v_k
  - Each consecutive pair v_i, v_{i+1} is joined by an edge in E
- Def. A path is ***simple*** if all nodes are *distinct*
- Def. An undirected graph is ***connected*** if ∀ pair of nodes u and v, there is a path between u and v



- *Short path*
- *Distance*

Cycles

- Def. A ***cycle*** is a path v_1, v_2, …, v_{k-1}, v_k in which $v_1 = v_k$, k > 3, and the first k-1 nodes are all distinct

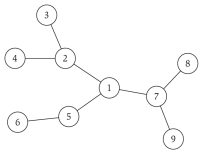


cycle C = 1-2-4-5-3-1

TREES

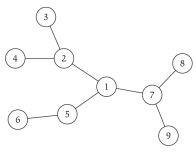
Trees

- Def. An undirected graph is a ***tree*** if it is connected and does not contain a cycle
- Simplest connected graph
  - Deleting any edge from a tree will disconnect it

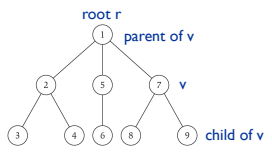


Rooted Trees

- Given a tree T, choose a root node *r* and orient each edge away from *r*
- Models hierarchical structure



a tree

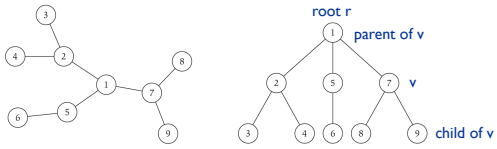


the same tree, rooted at 1

Why *n-1* edges?

Rooted Trees

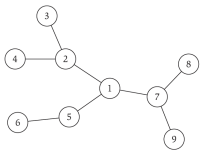
- Why *n-1* edges?
  - Each non-root node has an edge to its parent



a tree the same tree, rooted at 1

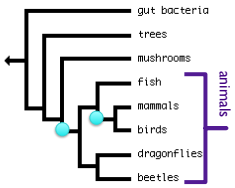
Trees

- Theorem. Let G be an undirected graph on *n* nodes. Any two of the following statements imply the third:
  - G is connected
  - G does not contain a cycle
  - G has *n*-1 edges



Phylogeny Trees

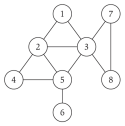
- Describe evolutionary history of species
  - mammals and birds share a common ancestor that they do not share with other species
  - all animals are descended from an ancestor not shared with mushrooms, trees, and bacteria



GRAPH CONNECTIVITY & TRAVERSAL

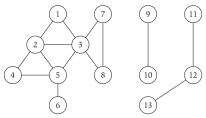
Connectivity

- **s-t connectivity problem.** Given nodes *s* and *t*, is there a path between *s* and *t*?
- **s-t shortest path problem.** Given nodes *s* and *t*, what is the length of the shortest path between *s* and *t*?
- Applications
  - Facebook
  - Maze traversal
  - Kevin Bacon number
  - Spidering the web
  - Fewest number of hops in a communication network



Application: Connected Component

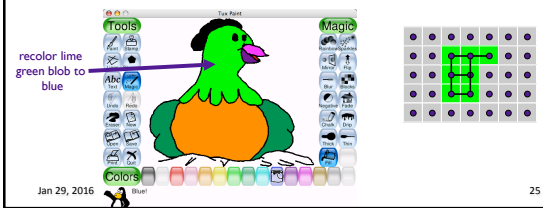
- Find all nodes ***reachable*** from *s*



- Connected component containing node 1 is { 1, 2, 3, 4, 5, 6, 7, 8 }

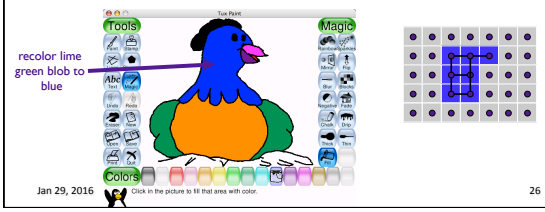
Application: Flood Fill

- Given lime green pixel in an image, change color of entire blob of neighboring lime pixels to blue
  - Node: pixel
  - Edge: two neighboring lime pixels
  - Blob: connected component of lime pixels

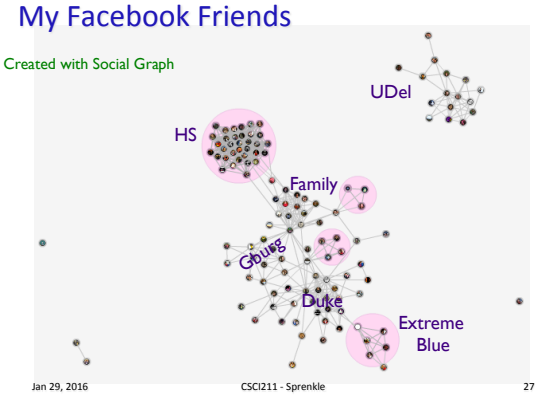


Application: Flood Fill

- Given lime green pixel in an image, change color of entire blob of neighboring lime pixels to blue
  - Node: pixel
  - Edge: two neighboring lime pixels
  - Blob: connected component of lime pixels

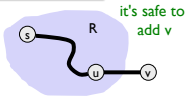


My Facebook Friends



A General Algorithm

```
R will consist of nodes to which s has a path
R = {s}
while there is an edge (u,v) where u∈R and v∉R
      add v to R
```



- *R* will be the **connected component** containing s
- Algorithm is underspecified

In what order should we consider the edges?

Possible Orders

- Breadth-first
- Depth-first

Looking Ahead

- Monday: journal - Chapter 2.5, 3-3.2
- Wednesday: Problem Set 3 due
  - Shorter problem set to review our recent material
- Wednesday: Midterm 1 handed out
  - Due following Wednesday at 5 p.m. (before Mock Con)
  - Work period – Monday before Mock Con
  - Midterm: your book, your notes, my lecture slides, me
    - No internet or other outside resources