CS 312: Algorithms

Dan Sheldon

Mount Holyoke College

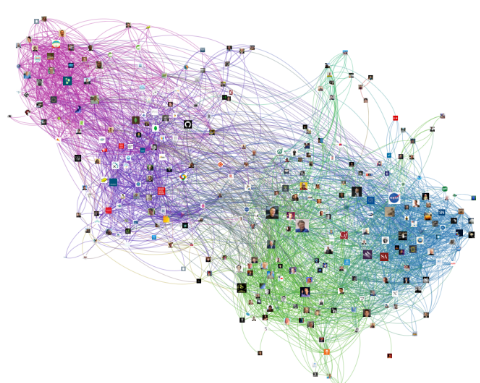
Last Compiled: September 19, 2018

Graphs: Motivation

- Facebook: how many "degrees of separation" between me and Barack Obama?
- Google Maps: what is the shortest driving route from Northampton to Florida?

How do we build algorithms to answer these questions?
Graphs and graph algorithms.

Networks



Networks

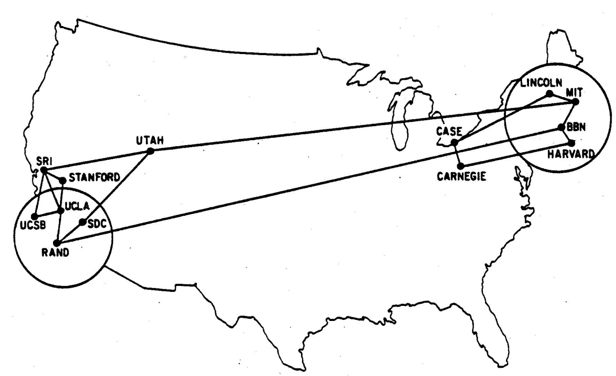
Graphs

A graph is a mathematical representation of a network

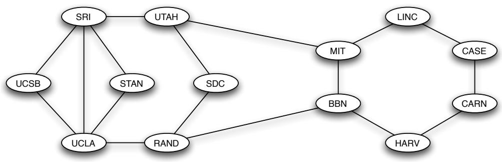
- Set of nodes (vertices) V
- Set of pairs of nodes (edges) E

Graph $G = (V, E)$

Example: Internet in 1970



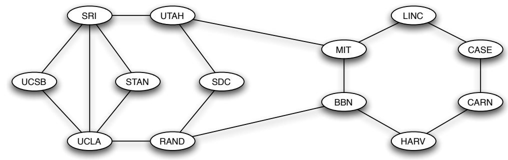
Example: Internet in 1970



Definitions:

Edge $e = \{u, v\}$ — but usually written $e = (u, v)$
u and v are *neighbors*, *endpoints* of e

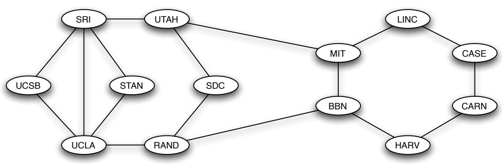
Example: Internet in 1970



Definitions:
A **path** is a sequence $P = v_1, v_2, \dots, v_{k-1}, v_k$ such that each consecutive pair v_i, v_{i+1} is joined by an edge in G

Path "from v_1 to v_k". A v_1–v_k path

Example: Internet in 1970

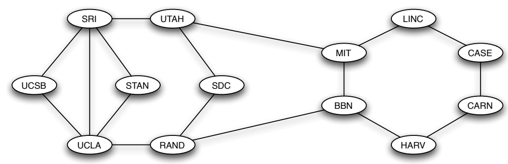


Definitions:

Q: Which is not a path?

1. UCSB - SRI - UTAH
2. LINC - MIT - LINC - CASE
3. UCSB - SRI - STAN - UCLA - UCSB
4. None of the above

Example: Internet in 1970

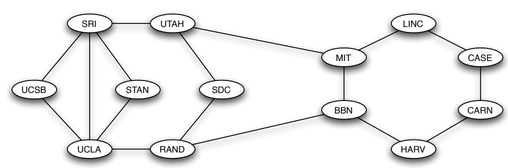


Definitions: Simple path, cycle, distance

Definitions

- **Simple path**: path where all vertices are distinct
  - HW exercise. Prove: If there is a path from u to v then there is a simple path from u to v.
- **Cycle**: path $v_1, \dots, v_{k-1}, v_k$ where
  - $v_1 = v_k$
  - First $k - 1$ nodes distinct
  - All edges distinct $(k > 3)$
- **Distance** from u to v: minimum number of edges in a u–v path

Example: Internet in 1970



Definitions:

Connected graph = graph with paths between every pair of vertices.

Connected component?

Definitions

- **Connected component**: maximal subset of nodes such that a path exists between each pair in the set
- maximal = if a new node is added to the set, there will no longer be a path between each pair

Definitions

- **Tree**: connected graph with no cycles
- Q: Is this equivalent to trees you saw in Data Structures?
- A: More or less.
- **Rooted tree**: tree with parent-child relationship
  - Pick root r and "orient" all edges away from root
  - Parent of v = predecessor on path from r to v

Directed Graphs

Graphs can be *directed*, which means that edges point *from* one node *to* another, to encode an asymmetric relationship. We'll talk more about directed graphs later.

Graphs are *undirected* if not otherwise specified.

Graph Traversal

Thought experiment. World social graph.

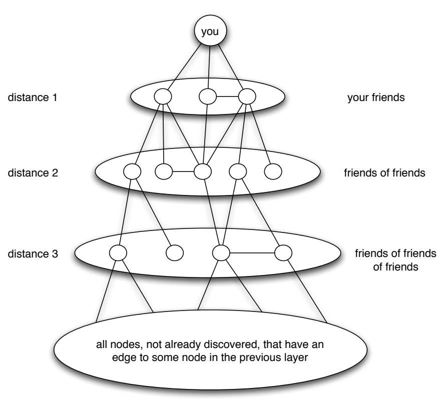
- Is it connected?
- If not, how big is largest connected component?
- Is there a path between you and Barack Obama?

How can you tell algorithmically?

Answer: graph traversal! (BFS/DFS)

Breadth-First Search

Explore outward from starting node s by distance. "Expanding wave"



Breadth-First Search: Layers

Explore outward from starting node s.

Define layer L_i = all nodes at distance exactly i from s.

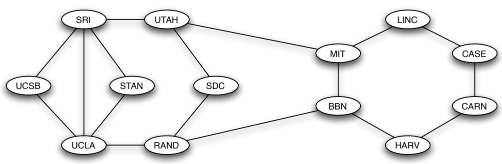
Layers

- $L_0 = \{s\}$
- L_1 = nodes with edge to L_0
- L_2 = nodes with an edge to L_1 that don't belong to L_0 or L_1
- ...
- L_{i+1} = nodes with an edge to L_i that don't belong to any earlier layer.

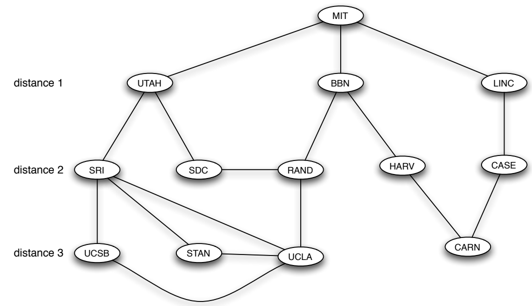
Observation: There is a path from s to t if and only if t appears in some layer.

BFS

Exercise: draw the BFS layers for a BFS starting from MIT

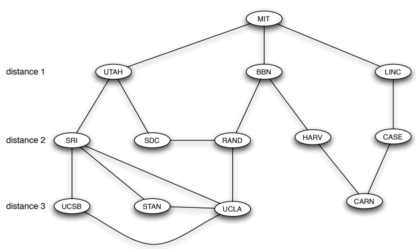


BFS Tree



We can use BFS to make a tree.

BFS Tree



Claim: let T be the tree discovered by BFS on graph $G = (V, E)$, and let (x, y) be any edge of G. Then the layer of x and y in T differ by at most 1.

Proof on board

BFS and non-tree edges

Claim: let T be the tree discovered by BFS on graph $G = (V, E)$, and let (x, y) be any edge of G. Then the layer of x and y in T differ by at most 1.

Proof

- Let (x, y) be an edge
- Suppose $x \in L_i$, $y \in L_j$, and $j > i + 1$
- When BFS visits x, either y is already discovered or not.
  - If y is already discovered, then $j \leq i + 1$. Contradiction.
  - Otherwise since $(x, y) \in E$, y is added to L_{i+1}. Contradiction.