Lecture 11

CSE 331

Sep 20, 2019

Mini Project group due week from Monday!

CSE 331 Mini project choices

Fall 2019

Please check the table below before submitting your mini project team composition to make sure your case study is not being used by another group. Case studies are assigned on a first come first serve basis.

Group	Chosen Algorithm	Case Study	Links
Daniel Shekhtman, William Nicholson, Andrew Quinonez (D's Get Degrees)	PageRank	Manipulation of PageRank for nefarious purposes	Link 1, Link 2, Link 3, Link 4
Jordan Clemons, Chris Burton, Christopher Perez (Group 1)	Pagerank	Google's use of Pagerank in sorting search results	Link 1, Link 2
Moulid Ahmed, Shrishty Shivani Jha, Shreya Lakhkar (ACE-MA)	Spotify Recommendation	Machine Learning Algorithm	Link 1, Link 2, Link 3
Justin Henderson, Hannah Wlasowicz, Judy Mei (PizzaTime)	Aes 256	ransomware	Link 1
Gillian Marcus, Jason Niu, Sharon Stack (2n^2 (//pls substitute caret for a superscript))	Deep Neural Networks for YT Recommendations	Social Media Targeted Advertising	Link 1, Link 2, Link 3, Link 4
Jiwon Choi, Matthew Ferrera, Winnie Zheng (The	Dijkstra's Algorithm	Maps/ Transportation Routes	Link 1, Link 2,

Please respect the TAs

note

stop following 131 views

Please be respectful to your TAs!

We always strive to be respectful of you during our interactions and I hope you return the courtesy to the TAs.This does not mean that you cannot ask TAs question and/or disagree with what they say but please do so in a civil and professional manner.

Please remember they are fellow students like you and they are really doing it for the love of helping y'all out and not for money (they can make more money doing other stuff). Many of you are UTAs, so you already know this but many of you might not (but now you do).

So if at any point of time you feel frustrated with an interaction with a TA, the correct thing to do is to come and talk with me (or the TA if you are comfortable doing that)-- lashing out at the TA is not done.

Also not done is any statement that demeans or degrades a TA: e.g. saying "I do not want to talk to the girl TA." *Note:* this did *not* happen in 331 but it *did* happen in another course this semester and that is not acceptable. If you do not see why, I'm happy to talk with you in person about this (please send me an email and setup a time to chat). As I mentioned above, the TAs are here to help y'all out of the goodness of their heart and they do not need to take any s**t from you.

Again, I'm not saying that this has happened in 331 but since a similar incident occurred in another course, I figured I should make my position clear.
#pin

logistics office_hours diversity

~ An instructor (Sanchit Batra) thinks this is a good note ~

edit · good note | 7

Updated 1 day ago by Atri Rudra

HW 3 is out!

Homework 3

Due by **11:00am, Friday, September 27, 2019**.

Make sure you follow all the homework policies.

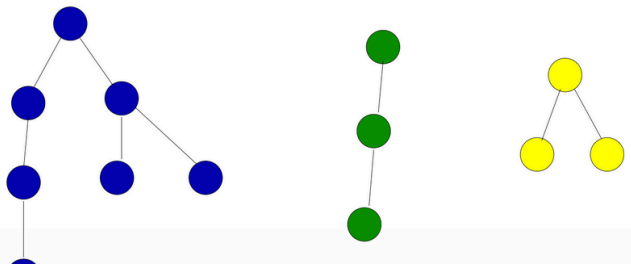
All submissions should be done via Autolab.

Sample Problem

The Problem

This problem is just to get you thinking about graphs and get more practice with proofs.

A **forest** with c *components* is a graph that is the union of c disjoint trees. The figure below shows for an example with $c = 3$ and $n = 13$ with the three connected components colored blue, read and yellow).



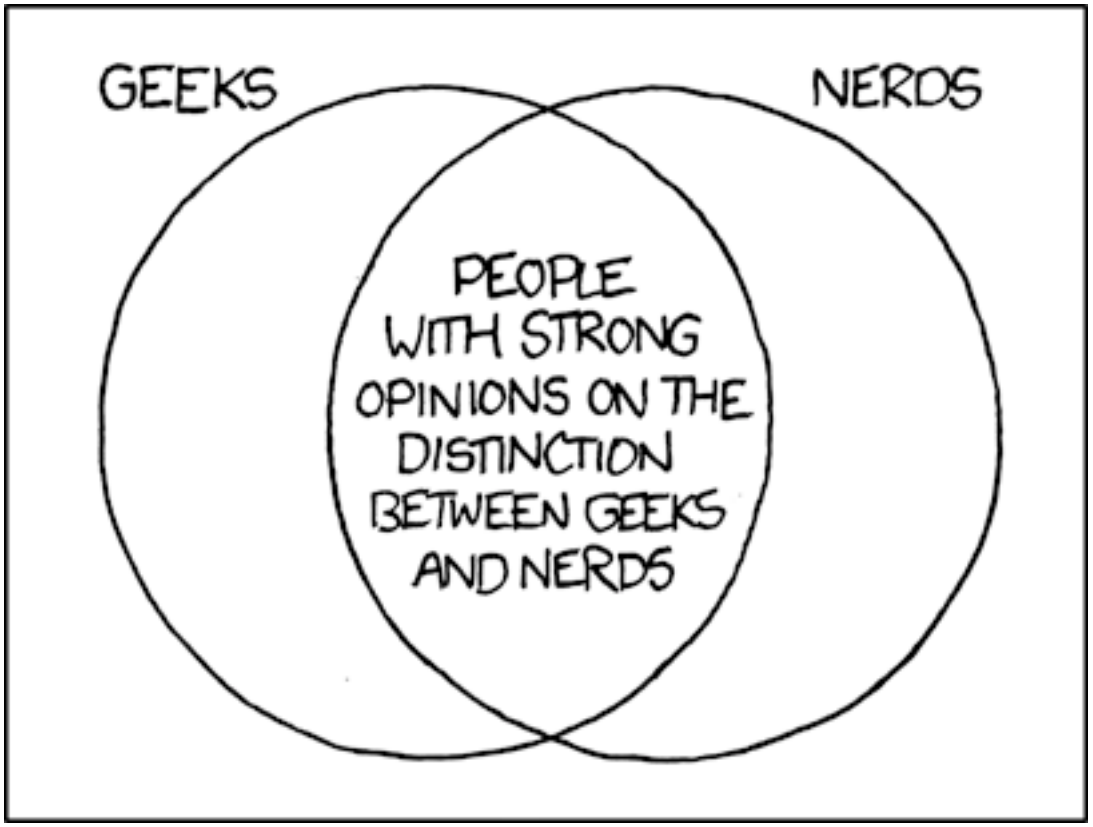
Solutions to HW 2

Handed out at the end of the lecture

Graded HW 1

Should be released by tonight

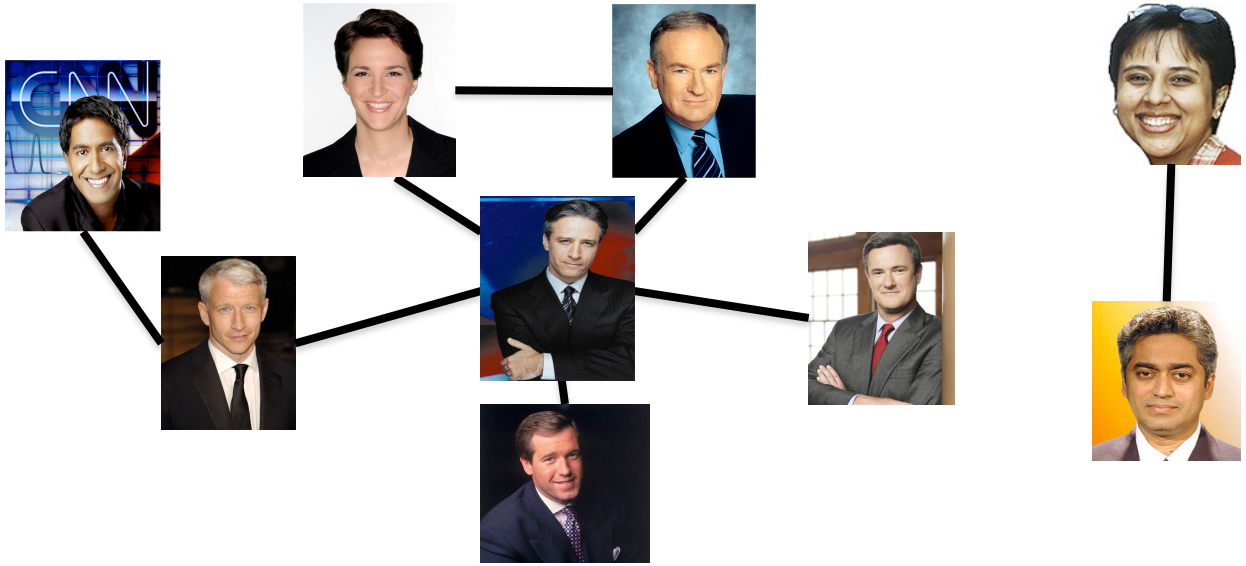
Formally define everything



http://imgs.xkcd.com/comics/geeks_and_nerds.png

Distance between u and v

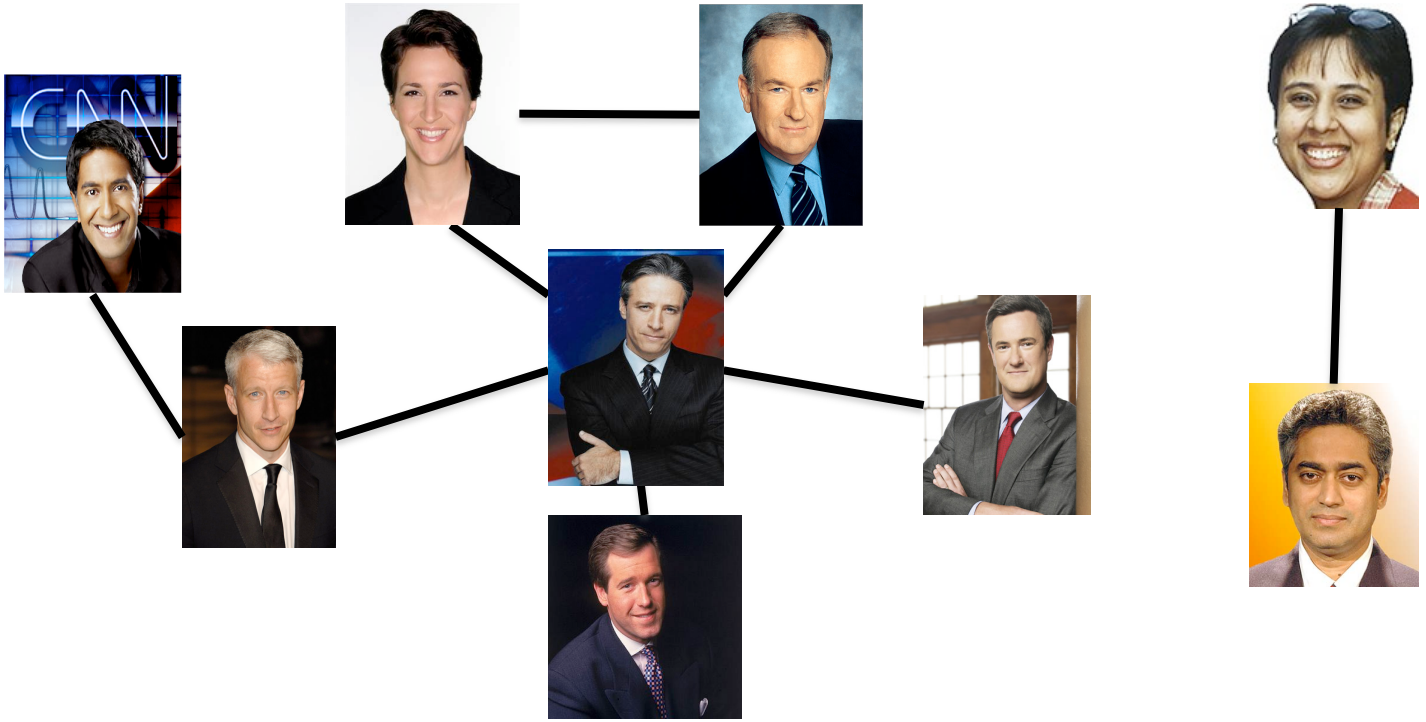
Length of the shortest length path between u and v



Distance between RM and BO? 1

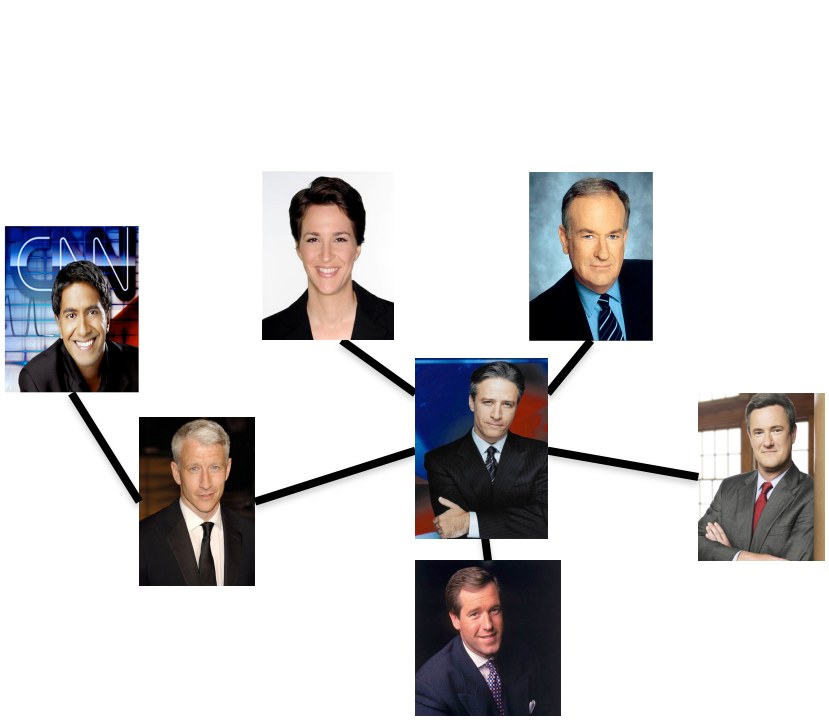
Tree

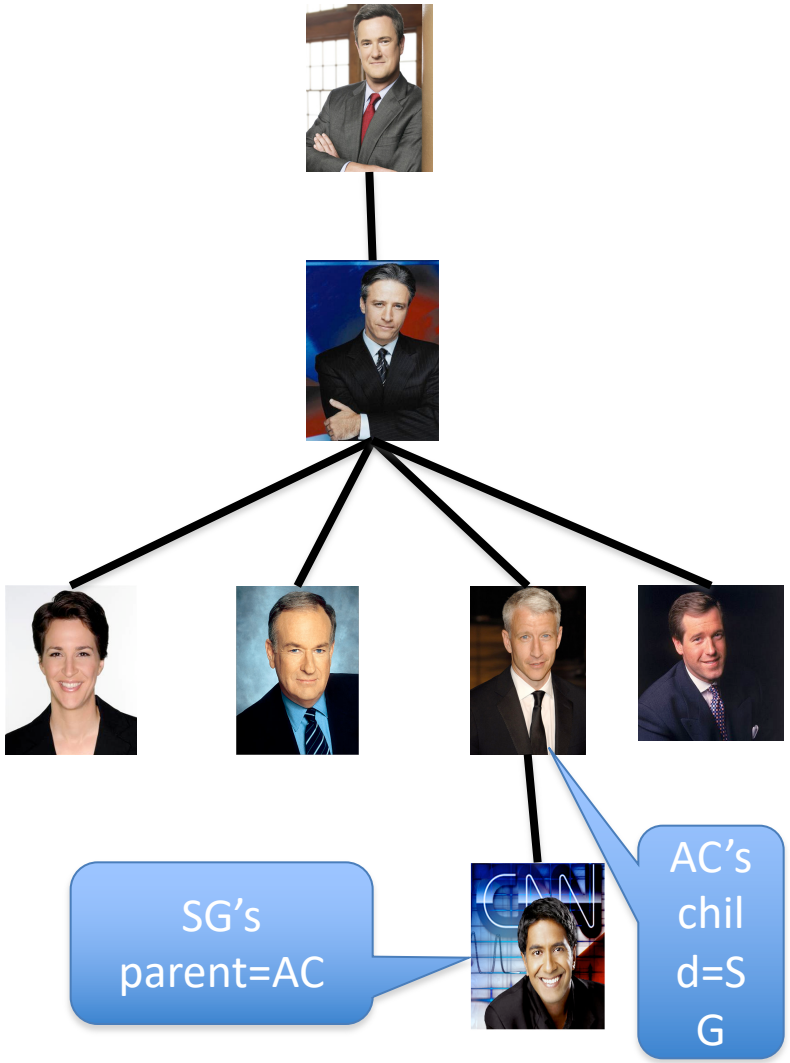
Connected undirected graph with no cycles



Rooted Tree

A rooted tree





Pick any vertex as root

Let the rest of the tree hang under "gravity"

Every n vertex tree has n-1 edges

Trees

This page collects material from previous incarnations of CSE 331 on trees, especially the proof that trees with n nodes have exactly $n - 1$ edges.

Where does the textbook talk about this?

`Section 3.1` in the textbook has the lowdown on trees.

Fall 2018 material

Here is the lecture video:

Every n vertex tree has n-1 edges

Let G be an undirected graph on n nodes

Then ANY two of the following implies the third:

T is connected

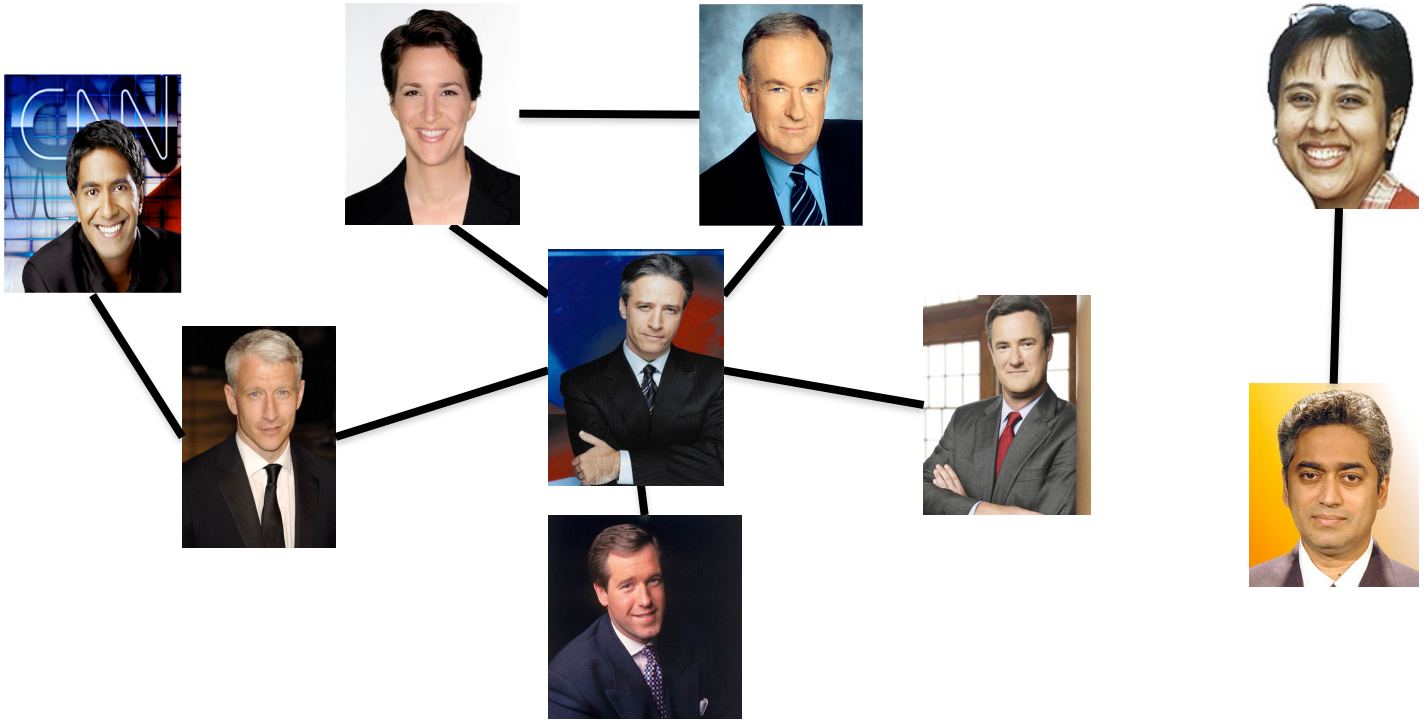
T has no cycles

T has n-1 edges

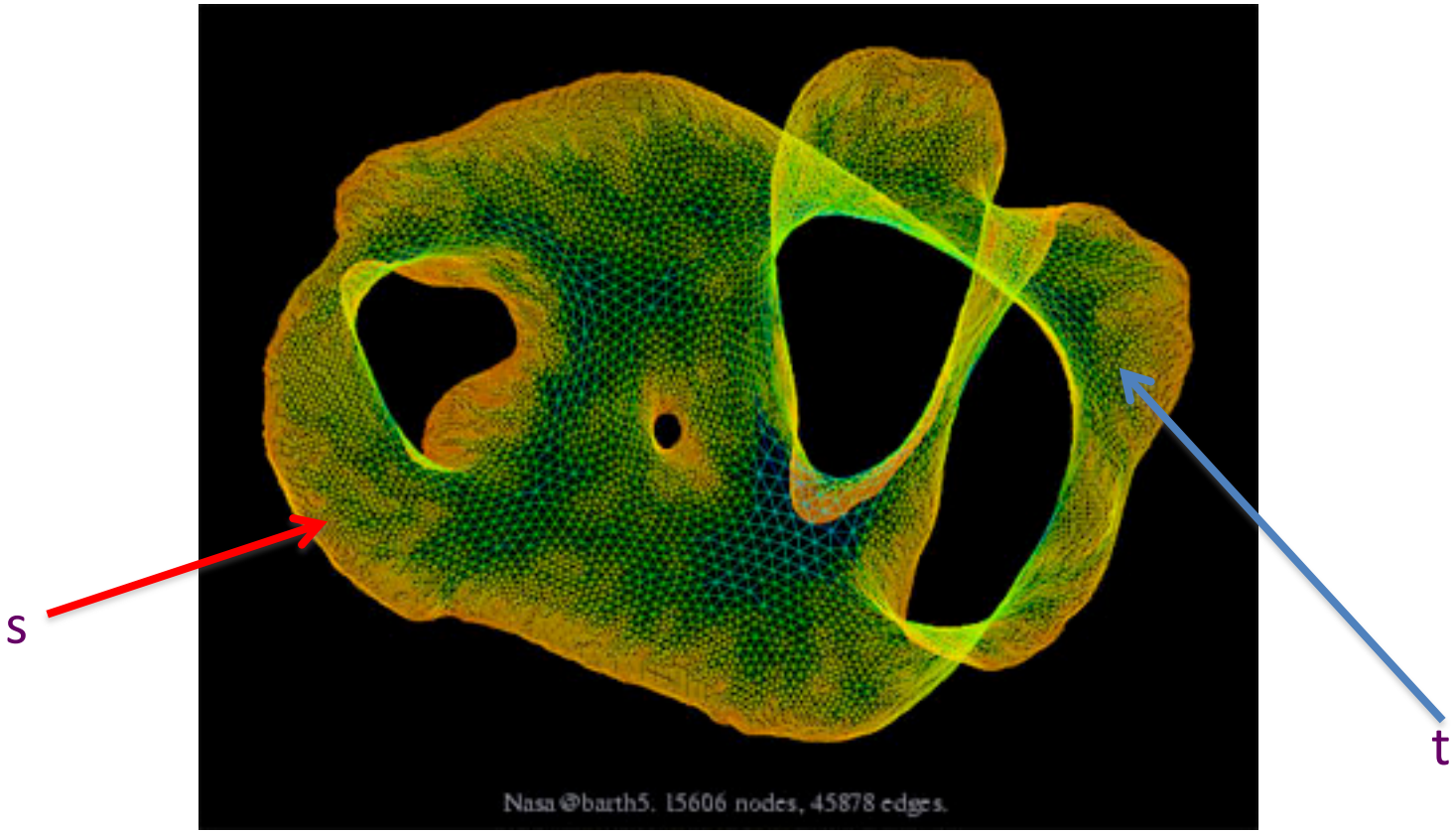
Rest of Today's agenda

Algorithms for checking connectivity

Checking by inspection



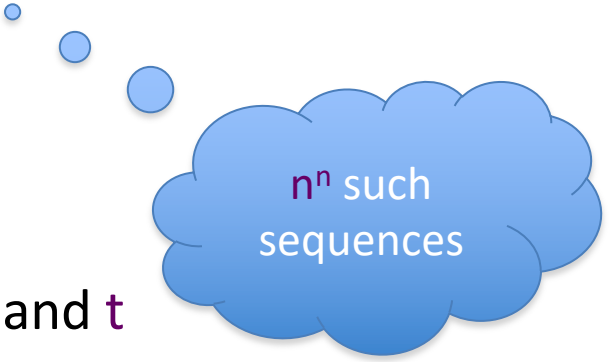
What about large graphs?



Are s and t connected?

Brute-force algorithm?

List all possible vertex sequences between s and t



Check if any is a path between s and t

Algorithm motivation

Breadth First Search (BFS)

BFS via examples

In which we derive the breadth first search (BFS) algorithm via a sequence of examples.

Expected background

These notes assume that you are familiar with the following:

- Graphs and their representation. In particular,
  - Notion of connectivity of nodes and connected components of graphs
  - Adjacency list representation of graphs
  - Notation:
    - $G = (V, E)$
    - $n = |V|$ and $m = |E|$
    - $CC(s)$ denotes the connected component of s
- Trees and their basic properties

The problem

In these notes we will solve the following problem:

Connectivity Problem

Input: Graph G = (V,E) and s in V

Output: All t connected to s in G

Breadth First Search (BFS)

Build layers of vertices connected to s

$L_0 = \{s\}$

Assume $L_0,..,L_j$ have been constructed

L_{j+1} set of vertices not chosen yet but are connected by an edge to L_j

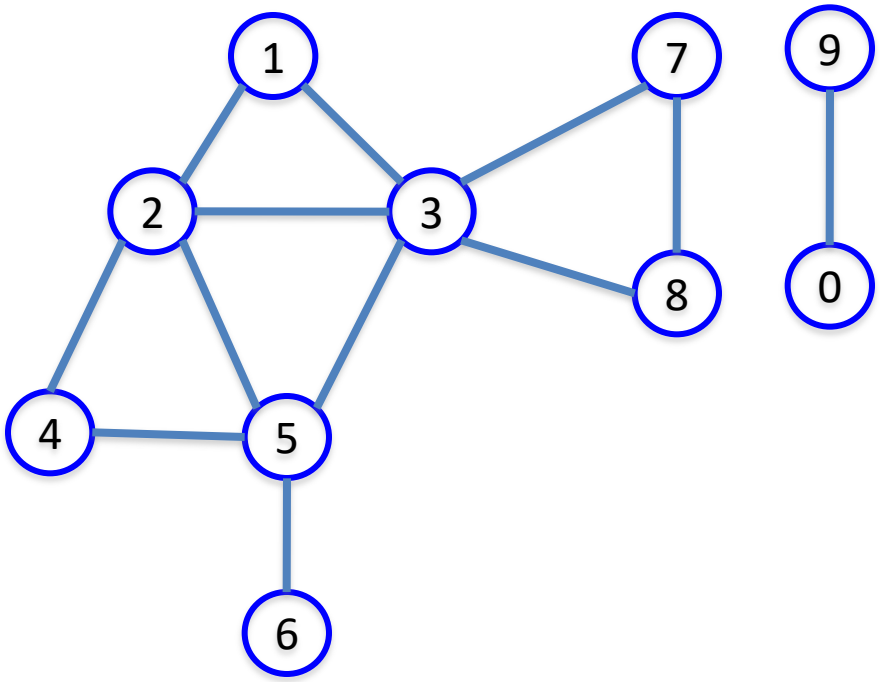
Stop when new layer is empty

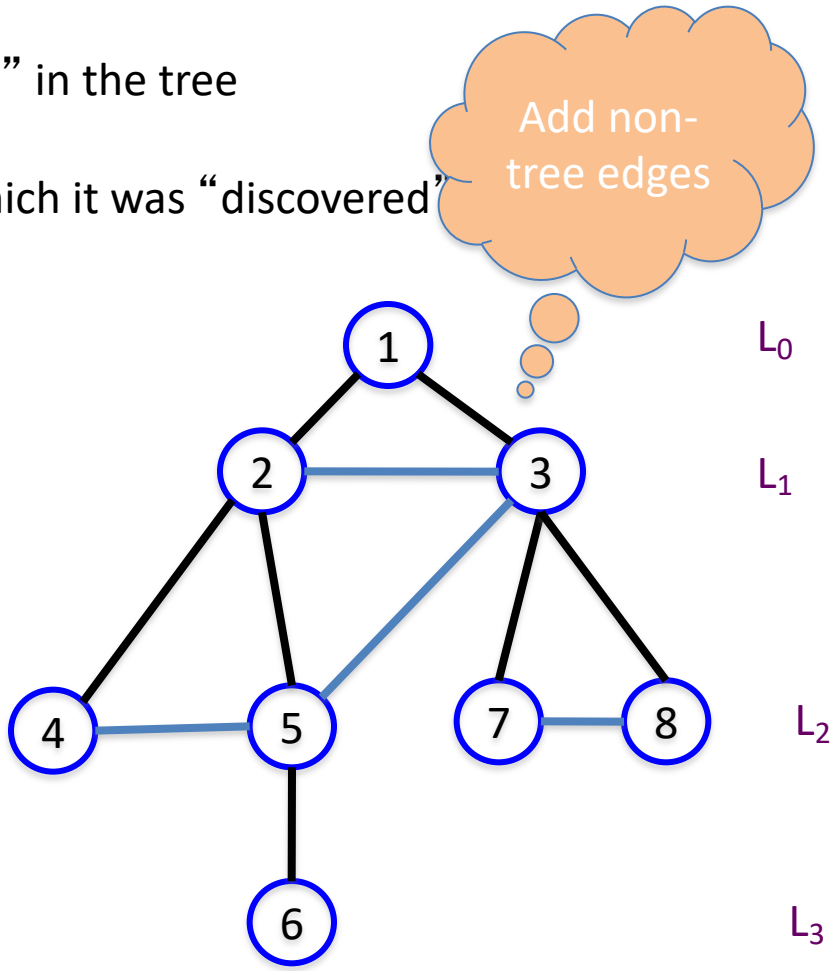
BFS Tree

BFS naturally defines a tree rooted at s

L_j forms the jth "level" in the tree

u in L_{j+1} is child of v in L_j from which it was "discovered"

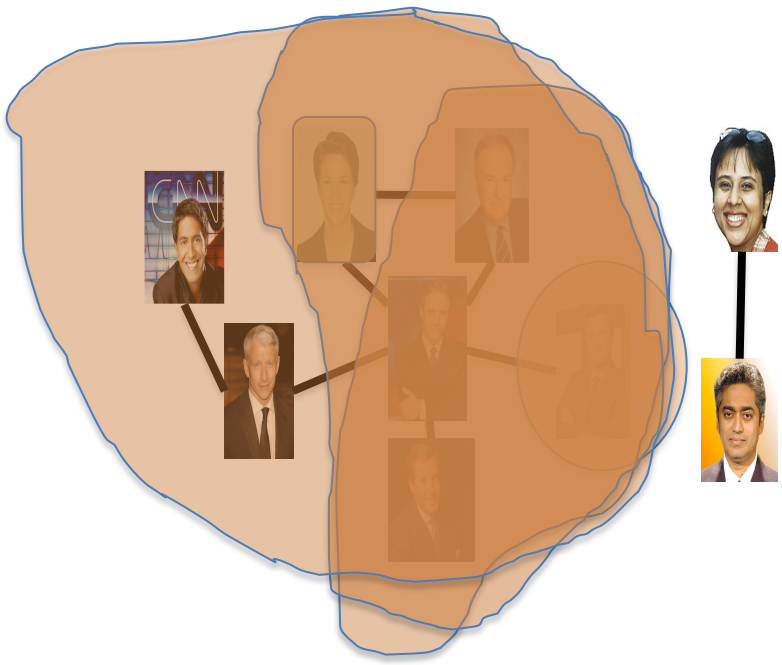




Today’s agenda

Computing Connected component

Computing Connected Component



Explore(s)

Start with R = {s}

While exists (u,v) edge v not in R and u in R

Add v to R

Output $R^* = R$

Questions?