Lecture 11

CSE 331

Sep 19, 2014

HW 2 due today

Place Q1, Q2 and Q3 in separate piles

I will not accept HWs after 1:15pm

Other HW related stuff

HW 3 has been posted online: see piazza

Solutions to HW 2 at the END of the lecture

Graded HW 1 available from Monday onwards

Out of town next week

Andrew Hughes will cover the lecture for me

note

stop following 58 views

Actions

Next Week

I will be out of town all of next week. (In case you're curious, I'll be here.)

I have arranged for a senior PhD student, Andrew Hughes (who will be teaching 396 next semester) to cover the lectures for me. The HWs will be handed out and collected as per the usual schedule. I will be checking piazza at least once per day when I'm out of town. The main thing that will change will be that I will not be hosting any office hours (online or otherwise). Frank and Zulkar will beef up their piazza participation for next week, so I'm confident all your questions will be answered in a timely manner.

If you have any questions and/or concerns, please let me know.

lectures homework3

edit good note | 0

1 day ago by Atri Rudra

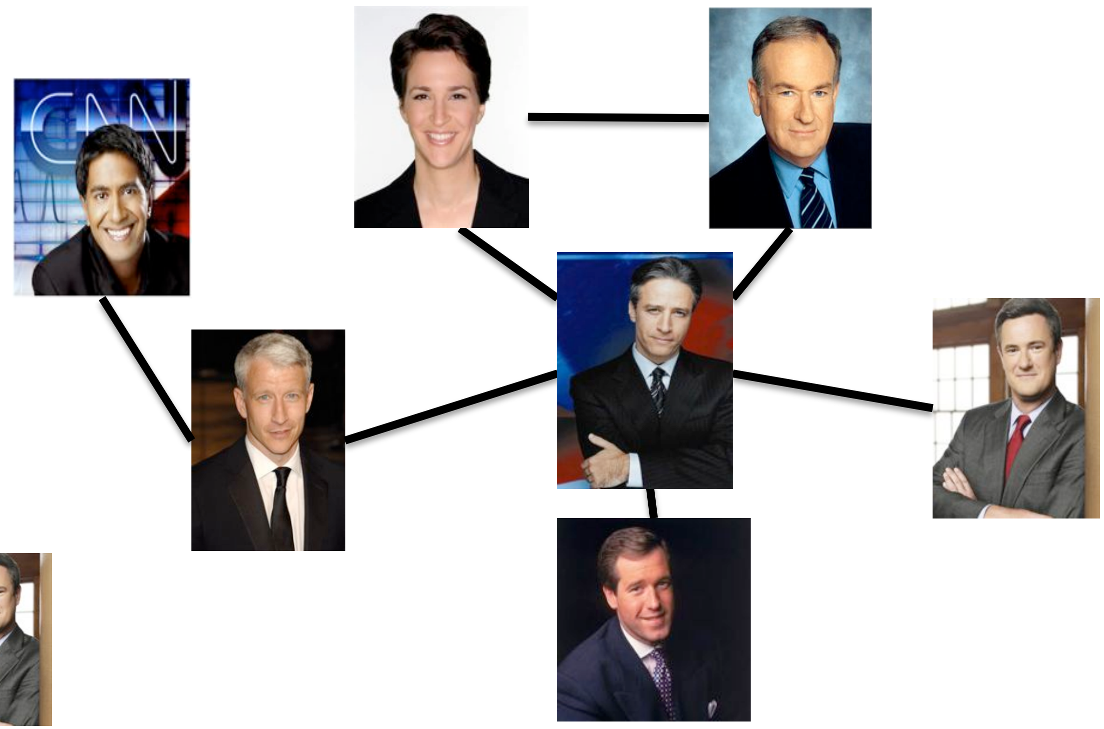
Graphs

Graph $G = (V,E)$

Directed vs Undirected (default)

No “self loops”

Paths



Connected

Sequence of vertices connected by edges

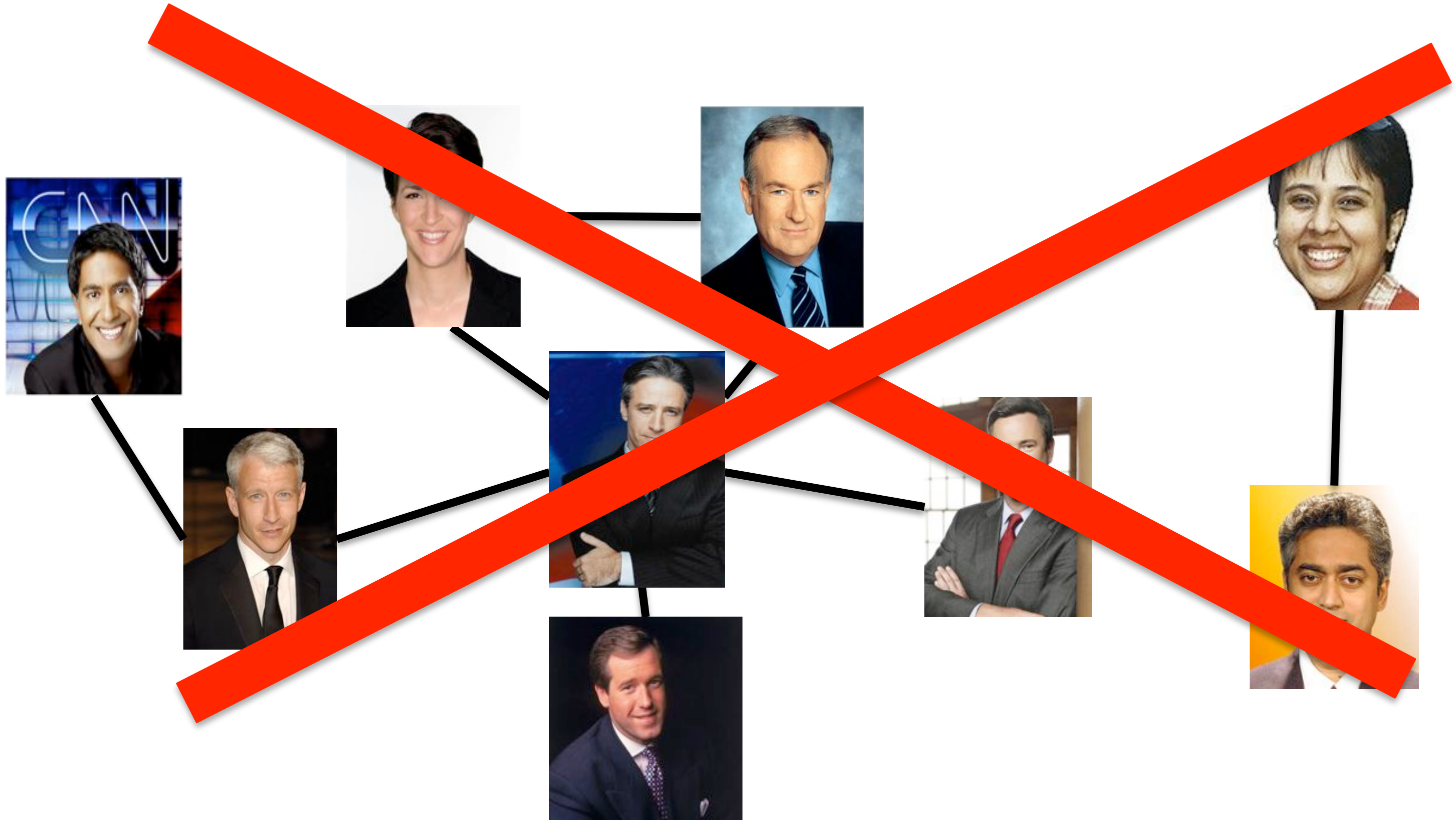
Path length 3

Connectivity

u and w are connected iff there is a path between them

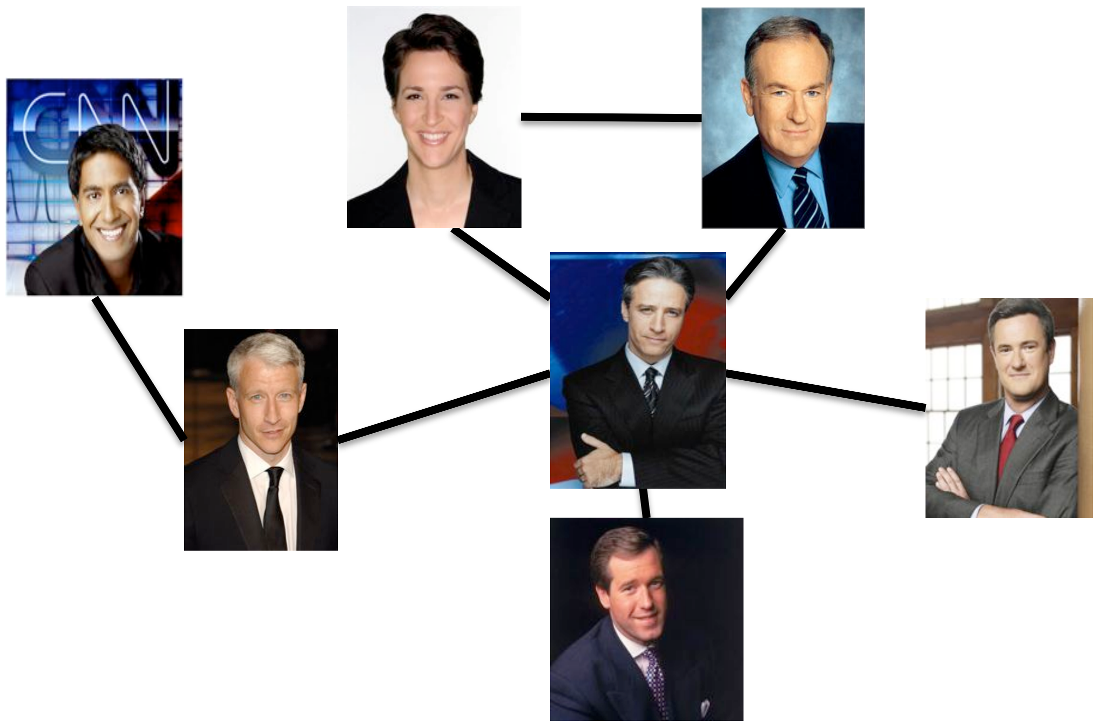
A graph is connected iff all pairs of vertices are connected

Connected Graphs



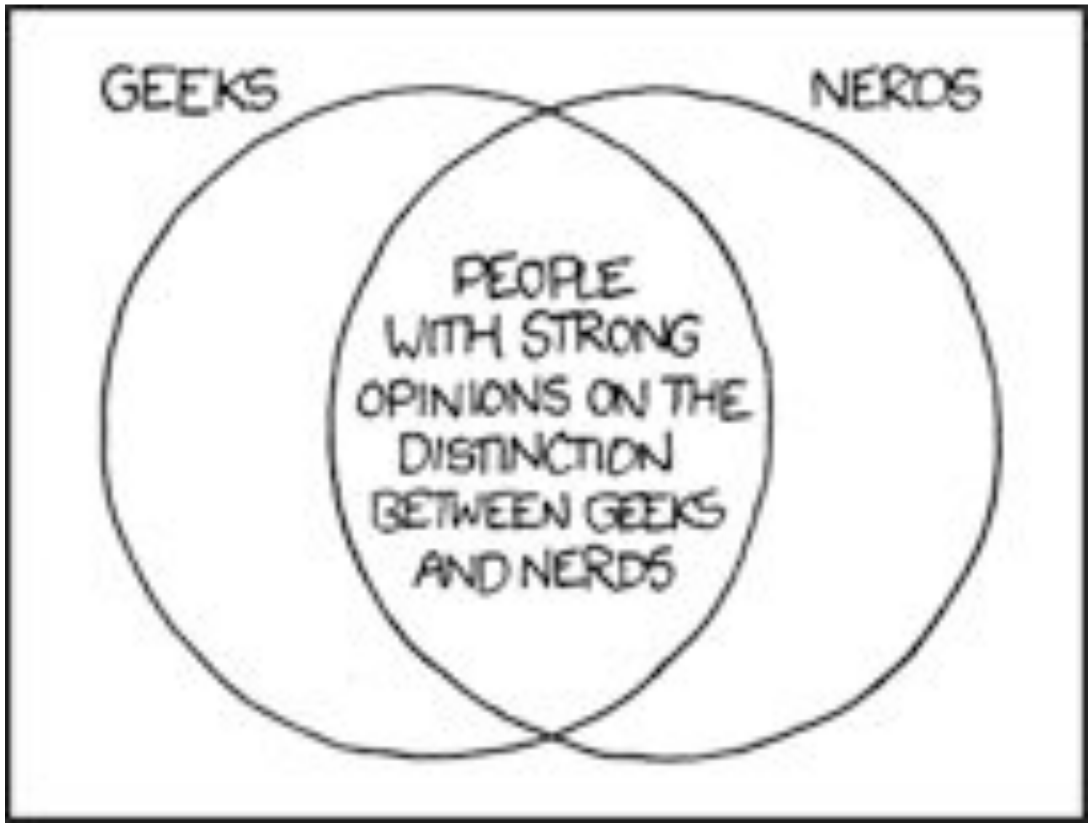
Every pair of vertices has a path between them

Cycles



Sequence of k vertices connected by edges, first k-1 are distinct

Formally define everything



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

Rest of Today's agenda

Formal definitions of paths, cycles, connectivity and trees

Prove n vertex tree has $n-1$ edges

Algorithms for checking connectivity

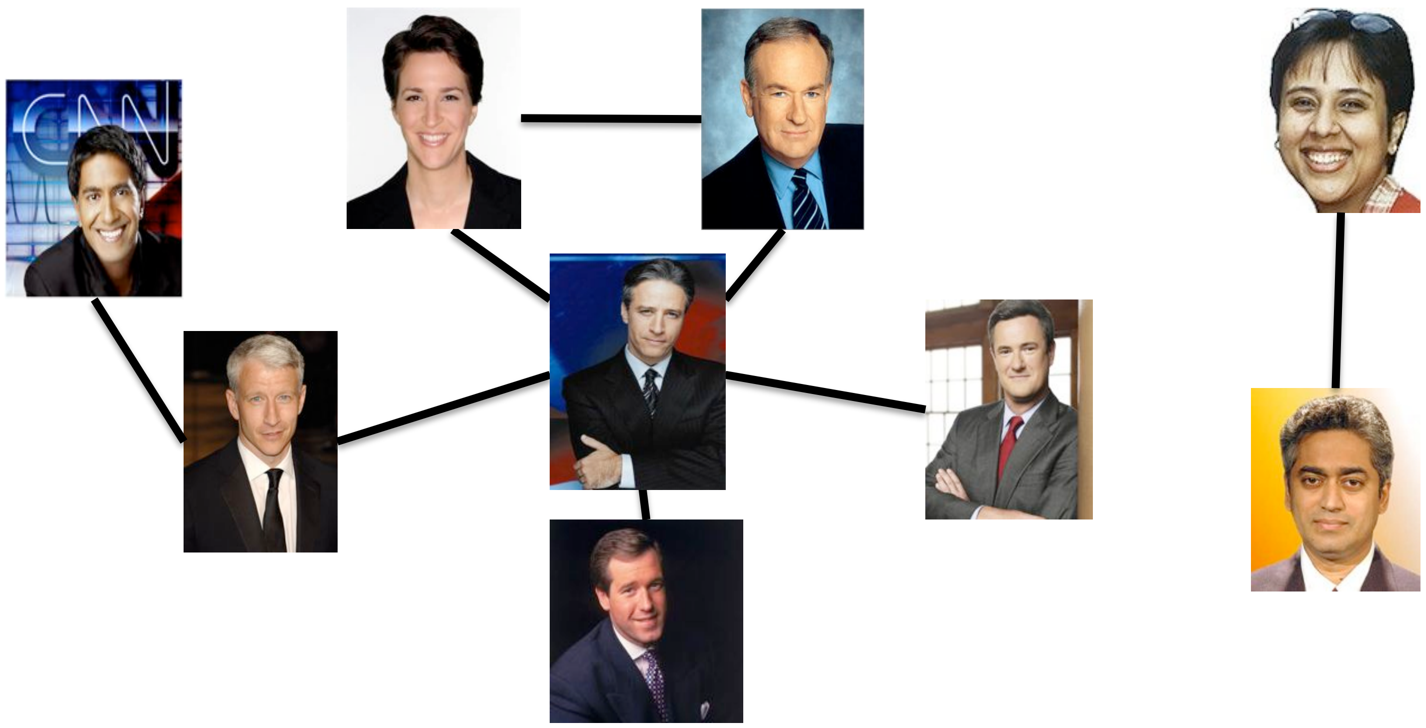
HW 2 due today

Place Q1, Q2 and Q3 in separate piles

I will not accept HWs after 1:15pm

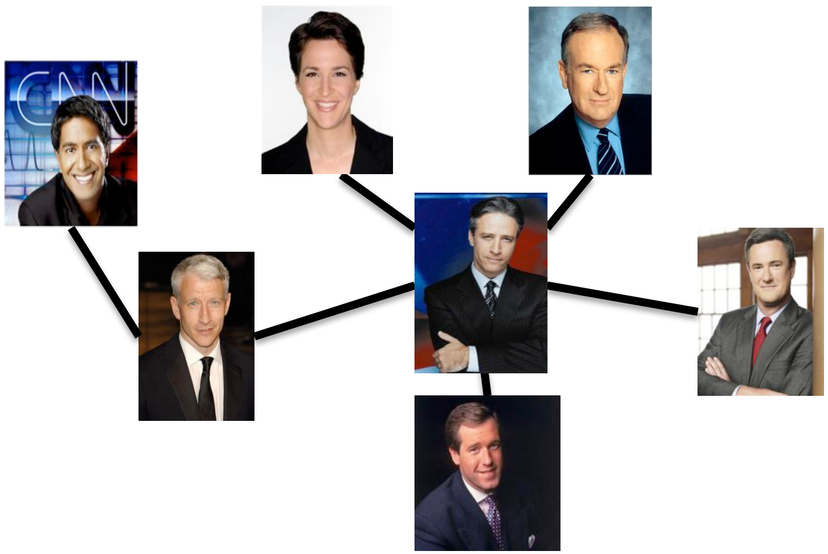
Tree

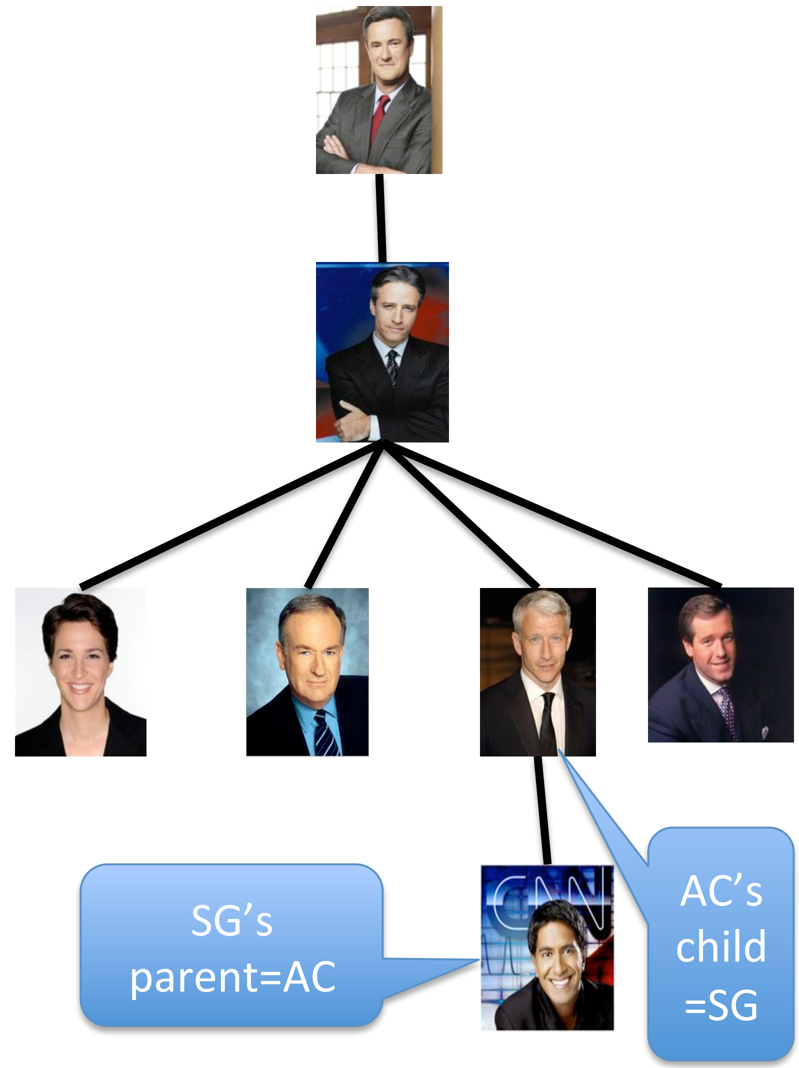
Connected undirected graph with no cycles



Rooted Tree

A rooted tree





Pick any vertex as root

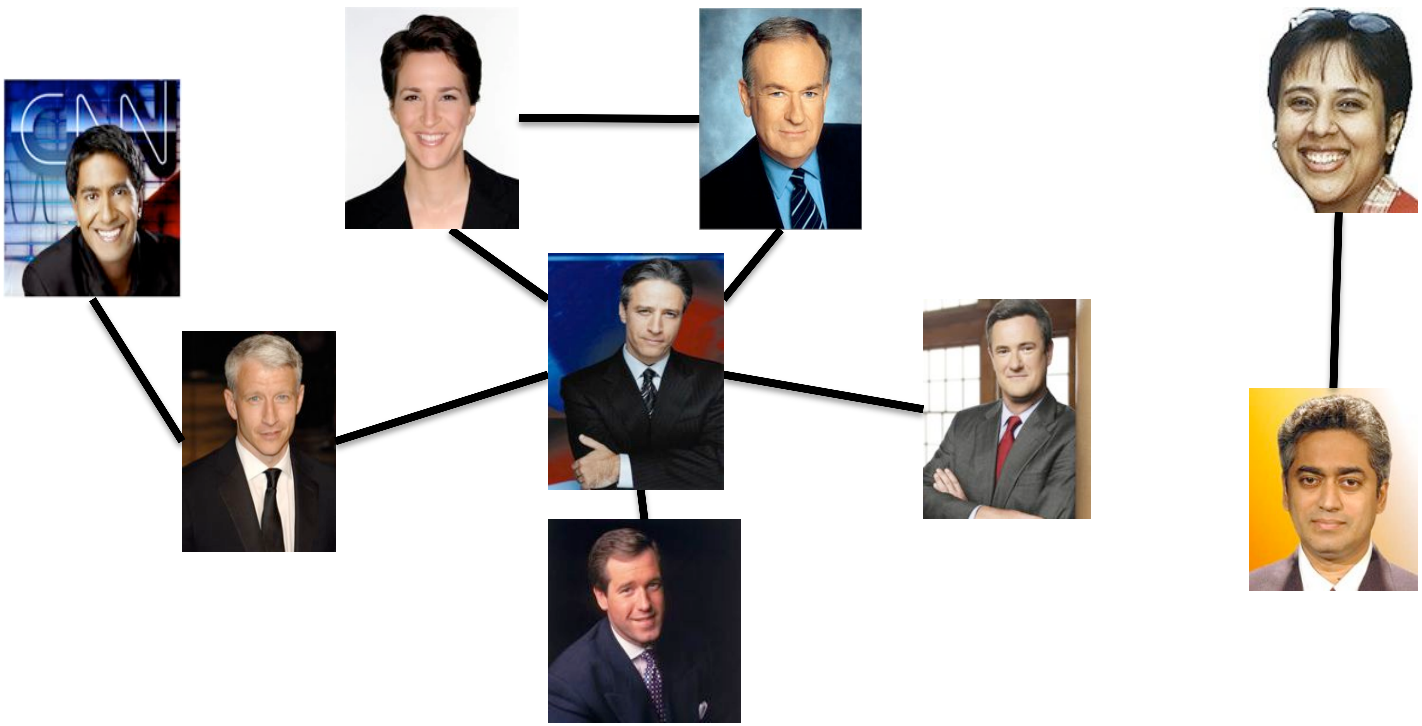
Let the rest of the tree hang under "gravity"

Rest of Today's agenda

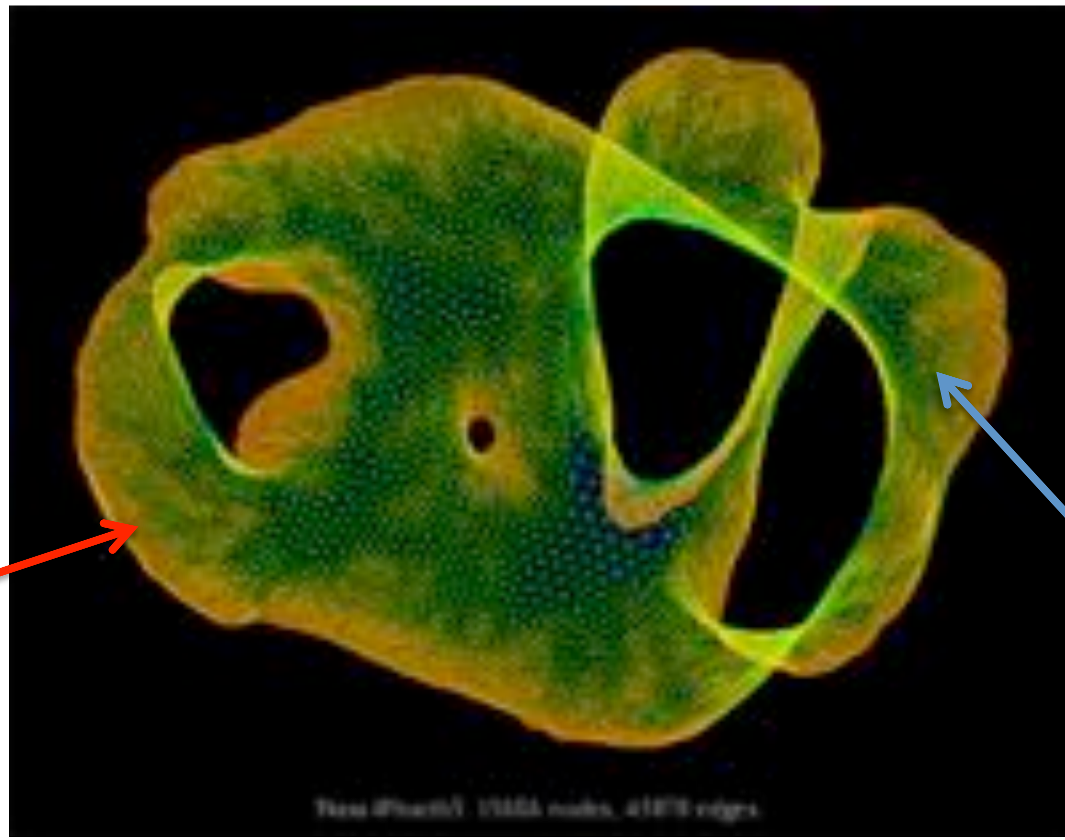
Prove n vertex tree has n-1 edges

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

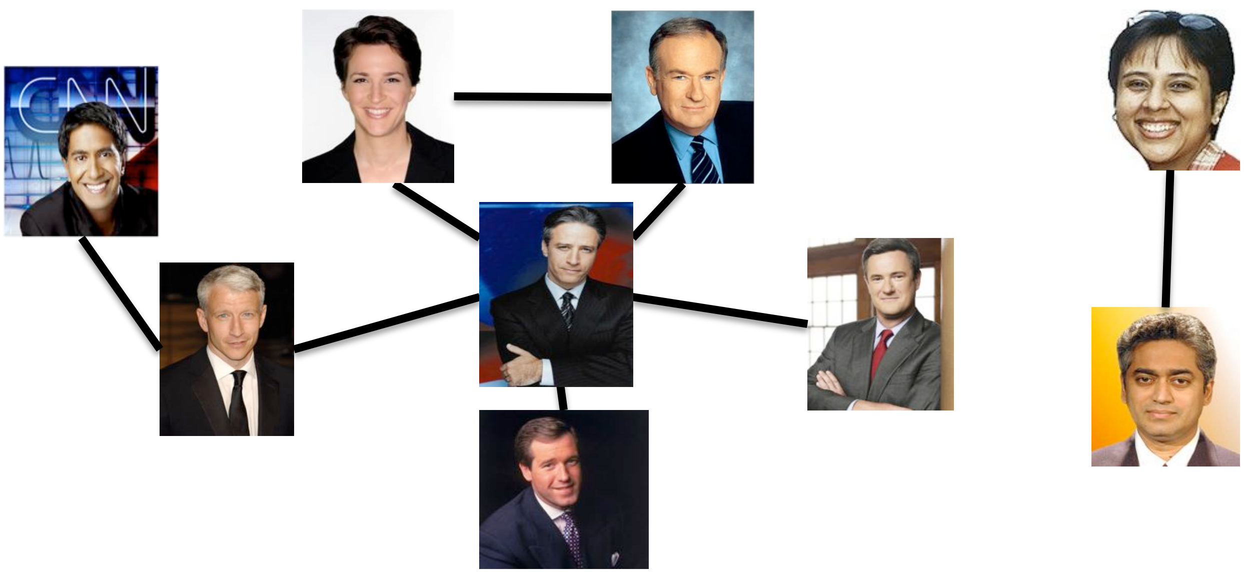
n^n such sequences

Check if any is a path between s and t

Algorithm motivation

Distance between u and v

Length of the shortest length path between u and v



Distance between RM and BO? 1

Questions?

Breadth First Search (BFS)

Is s connected to t?

Build layers of vertices connected to s

L_j : all nodes at distance j from s

$L_0 = \{s\}$

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

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

Stop when new layer is empty

Exercise for you

Prove that L_j has all nodes at distance j from s

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”