Lecture 13

CSE 331

Sep 26, 2018

Mini Project Video due Nov 5

CSE 331 Mini project choices

Fall 2018

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
Chinmayee Bandal, Sarah Peters, Tracy Zheng ()	Dijkstra's Algorithm	Google Maps	Link 1, Link 2
Jonathan Wong, Jacky Eng, Jack Bett (Segmentation Fault)	Linkedin Feed Algorithm	Linkedin Feed Relevance System	Link 1, Link 2, Link 3
Waiwai Kim, John Demetros, Frank Tsai (Autonomous Vehicle)	Deep Deterministic Policy Gradients	Reinforcement Learning in Autonomous Vehicle	Link 1, Link 2, Link 3
Mohammed Shmsuddin, Vencent Feng, Krapi Vani (Group 1)	Data Compression Algorithms	Sound data compression	Link 1
Nicholas Weiser, Matthew Lichtenthal, Vincent Bilotta (TryNotToFail)	Timber Match-Filtering Algorithm	Ancestry	Link 1, Link 2

Algorithms via examples closes in 5 day(s)

A total of **48** vote(s) in **33** hours

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 to L_j

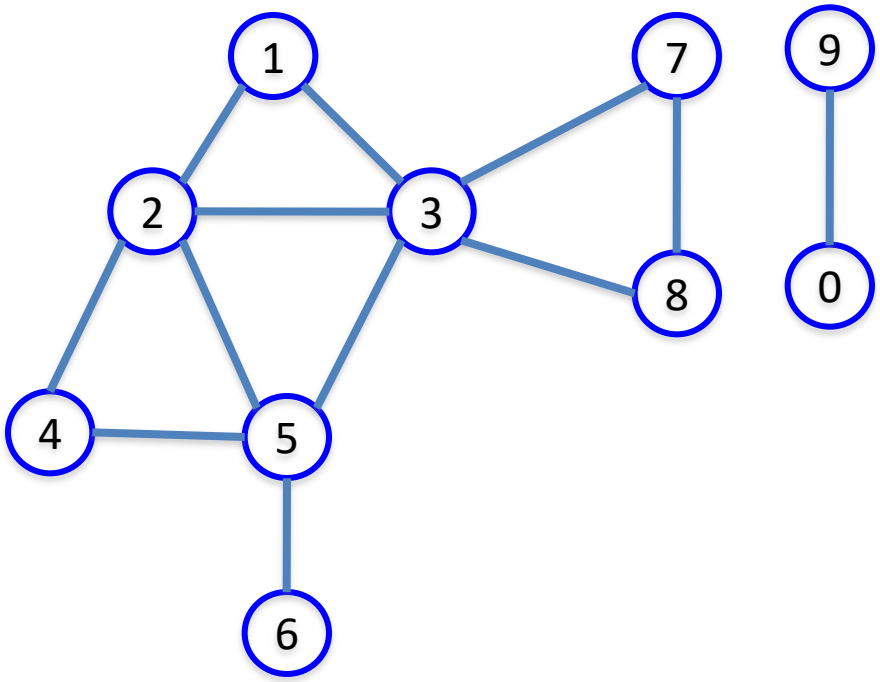
Stop when new layer is empty

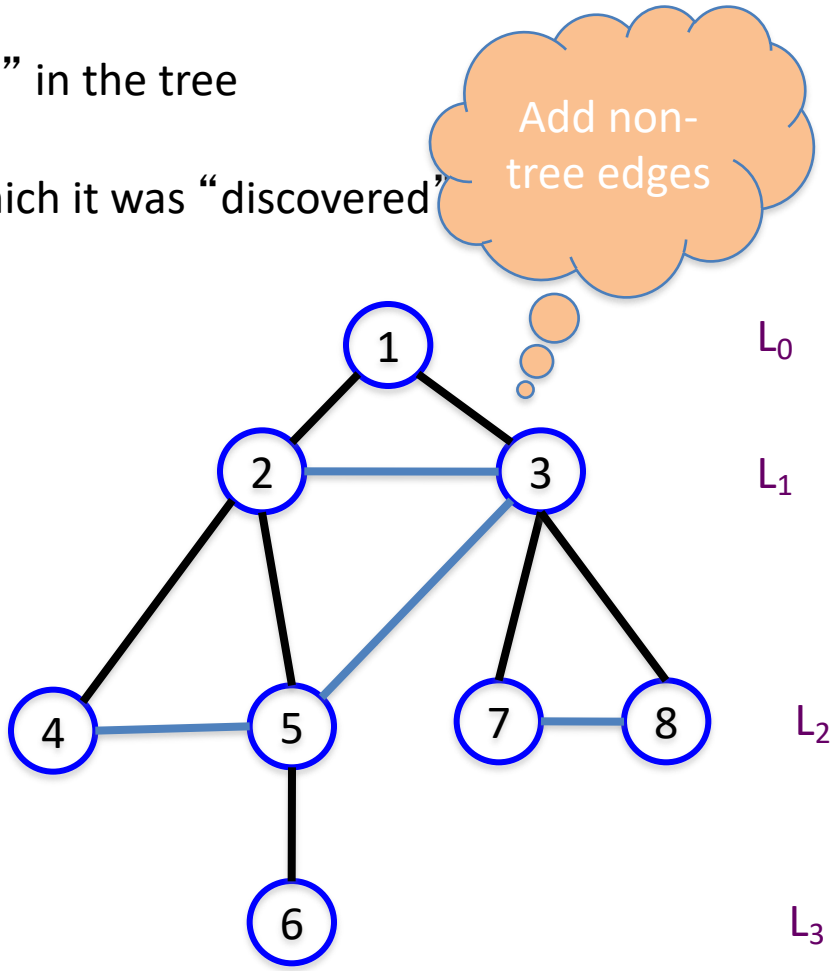
BFS Tree

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"

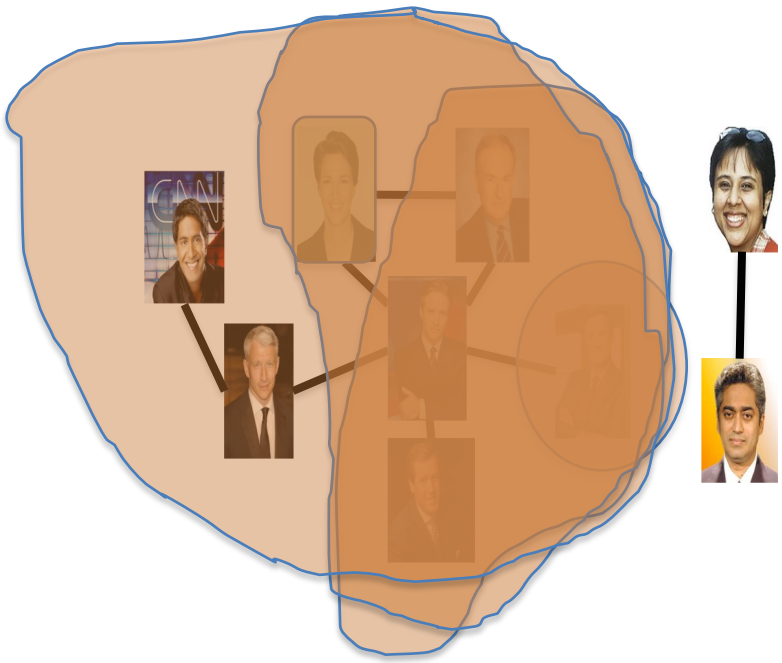




Today’s agenda

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

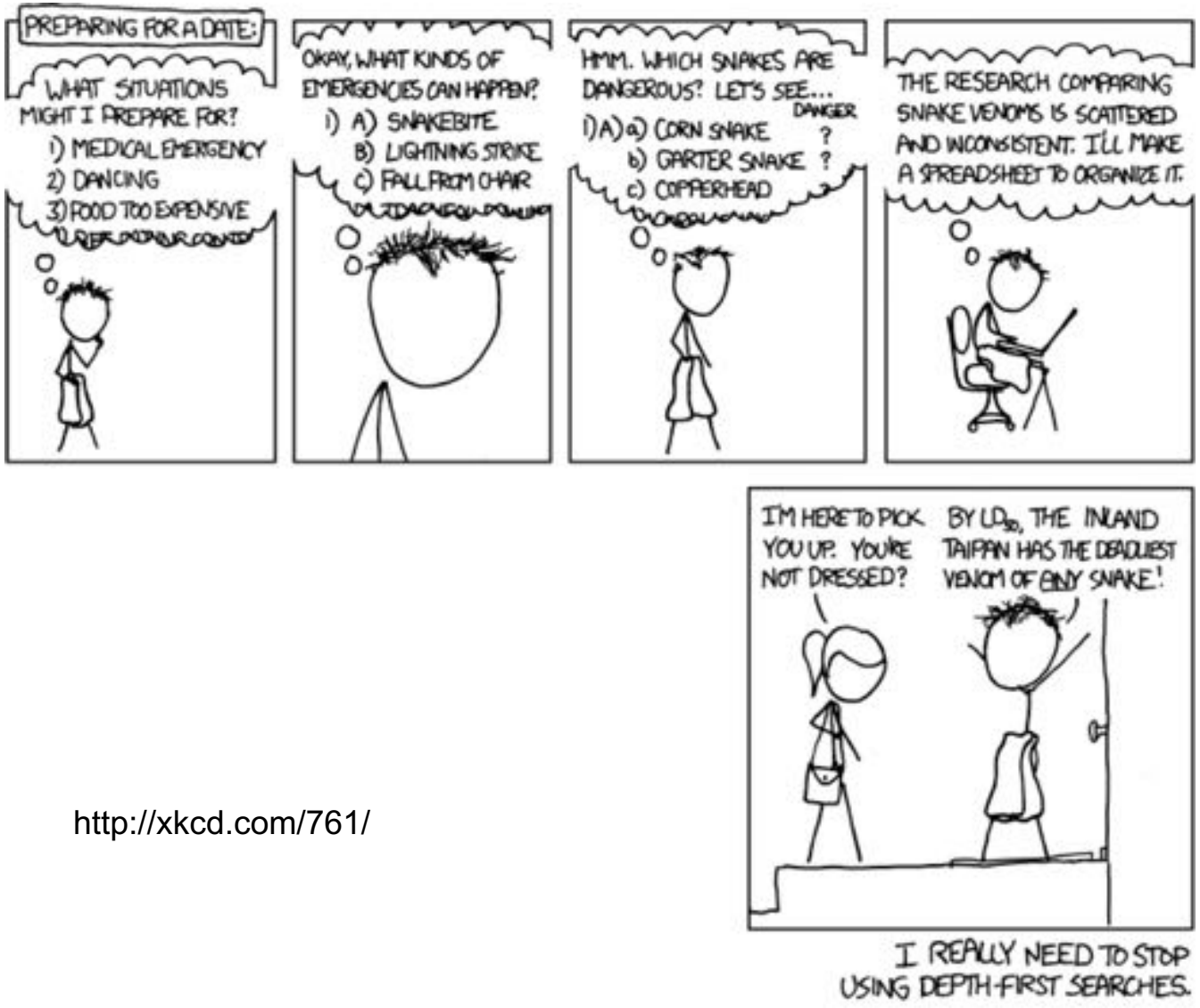
Output $R^* = R$

Questions?

BFS


© Original Artist
Reproduction rights obtainable from
www.CartoonStock.com
Studies have found that forwarding e-mail can end friendships. Please help spread the word by forwarding this to all friends.
search ID: mbcn800

Depth First Search (DFS)



http://xkcd.com/761/

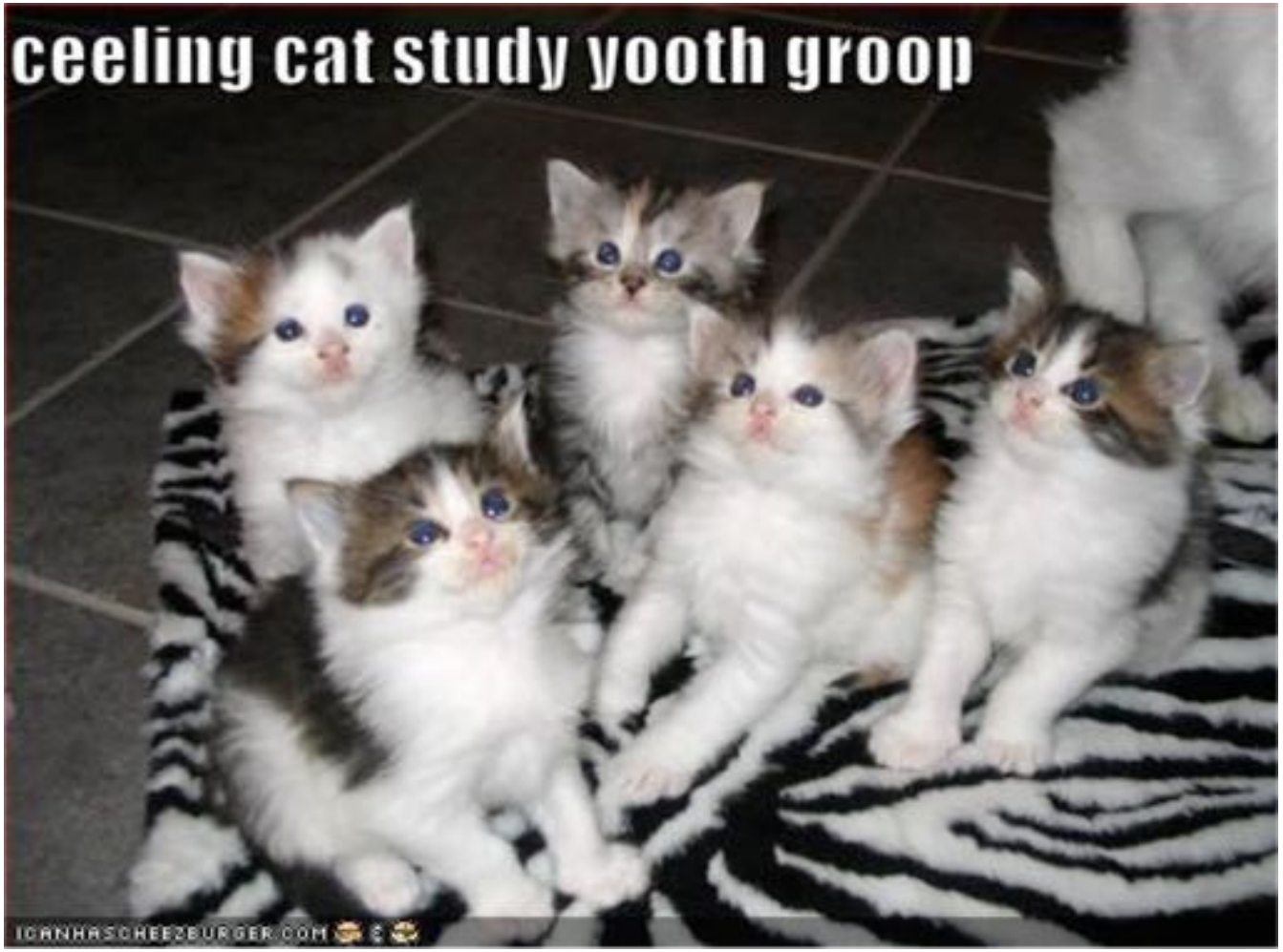
DFS(u)

Mark u as explored and add u to R

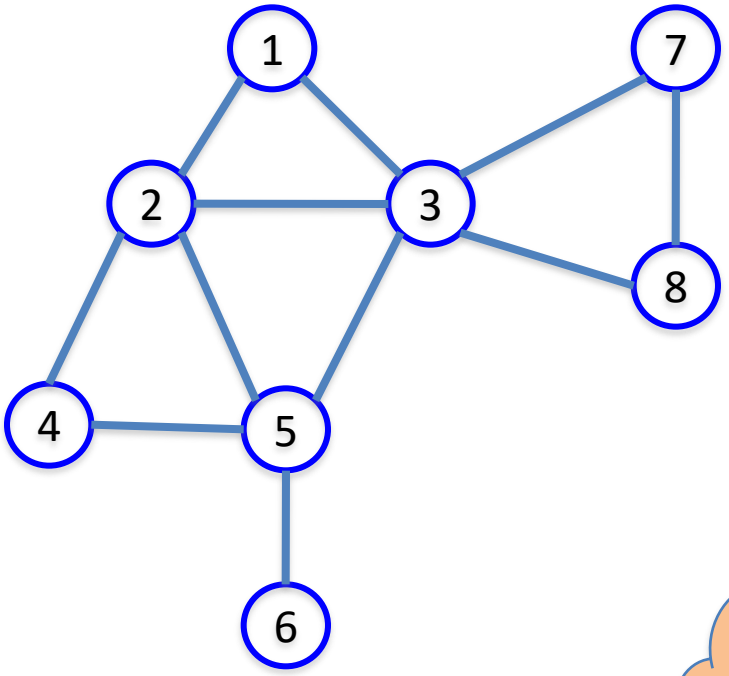
For each edge (u,v)

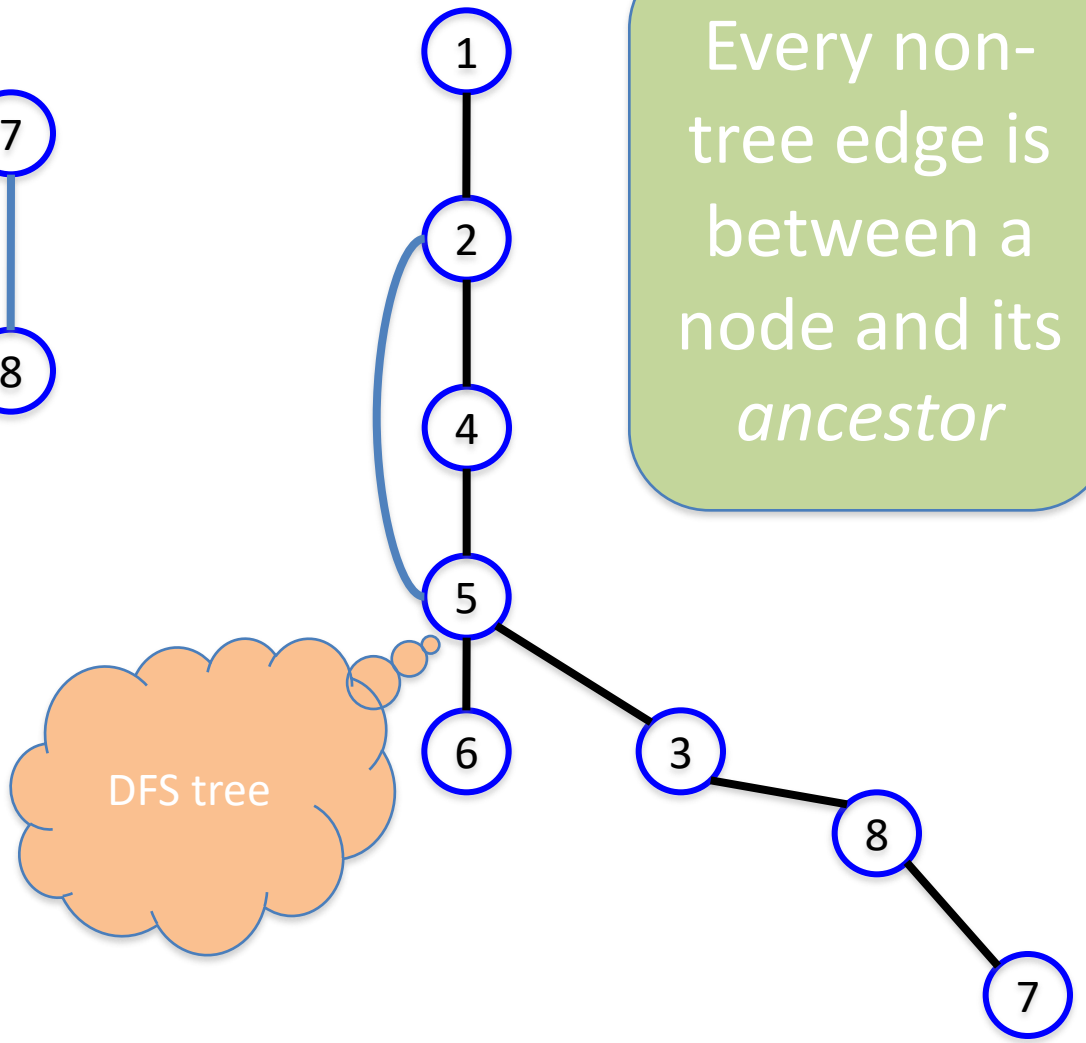
If v is not explored then DFS(v)

Why is DFS a special case of Explore?



A DFS run



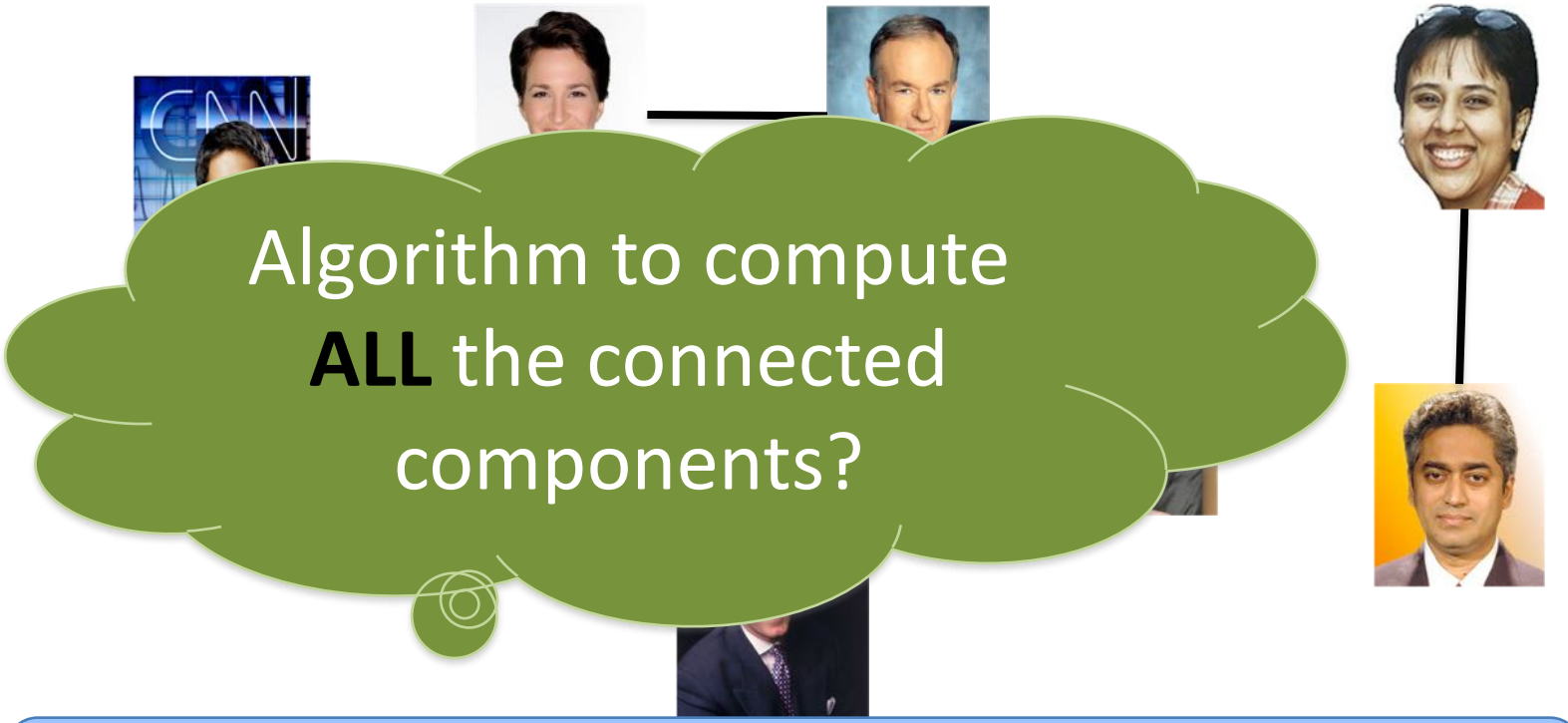


Every non-tree edge is between a node and its *ancestor*

Questions?

Connected components are disjoint

Either Connected components of s and t are the same or are disjoint



Run BFS on some node s. Then run BFS on t that is not connected to s