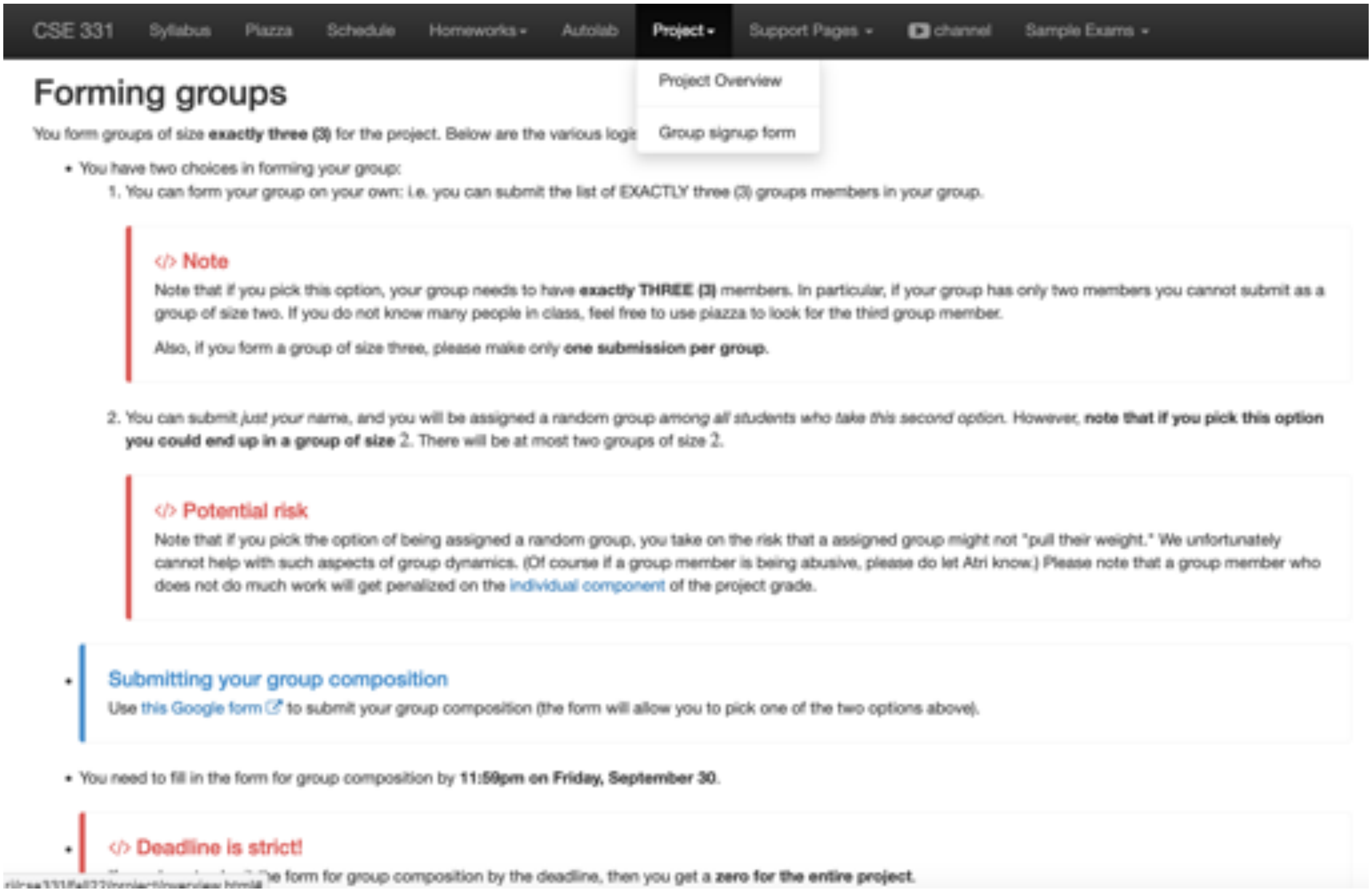
Lecture 11

CSE 331

Sep 23, 2022

Register project groups 1 week!

Deadline: Friday, Sep 30, 11:59pm



CSE 331 | Syllabus | Piazza | Schedule | Homeworks | Autolab | Project | Support Pages | channel | Sample Exams

Project Overview

Group signup form

Forming groups

You form groups of size **exactly three (3)** for the project. Below are the various logi

- You have two choices in forming your group:
  1. You can form your group on your own: i.e. you can submit the list of EXACTLY three (3) groups members in your group.

     </> Note

     Note that if you pick this option, your group needs to have **exactly THREE (3)** members. In particular, if your group has only two members you cannot submit as a group of size two. If you do not know many people in class, feel free to use piazza to look for the third group member.

     Also, if you form a group of size three, please make only **one submission per group**.

  2. You can submit *just your name*, and you will be assigned a random group *among all students who take this second option*. However, **note that if you pick this option you could end up in a group of size** 2. There will be at most two groups of size 2.

     </> Potential risk

     Note that if you pick the option of being assigned a random group, you take on the risk that a assigned group might not "pull their weight." We unfortunately cannot help with such aspects of group dynamics. (Of course if a group member is being abusive, please do let Atri know.) Please note that a group member who does not do much work will get penalized on the individual component of the project grade.

- Submitting your group composition

  Use this Google form to submit your group composition (the form will allow you to pick one of the two options above).

- You need to fill in the form for group composition by **11:59pm on Friday, September 30**.

- </> Deadline is strict!

  ...he form for group composition by the deadline, then you get a **zero for the entire project**.

If you need it, ask for help

Couple of clarifications

note @153 · stop following · 4 views

Actions

Couple of clarification pointers

- The first one is on references. If you are referring to an allowed source to cite a result, please make sure it is specific. So e.g. if you want to refer to result (1.6) in the textbook for correctness of GS, then explicitly state result (1.6). Or if you want to refer to recitation notes for week x, explicitly state you are referring to recitation notes from week x.
    - Note that you have to **explicitly** cite recitation notes-- putting recitation notes in your sources is not enough.
- Come clarifications on HW 1 solutions:
    - The solutions we hand out in class is essentially the "perfect" solution-- an upper bound on what will get you a level 5 if you will. It is however not a *lower bound* on what can get you a level 5. In other words, even if your solution does not look like the solutions (e.g. not as detailed as the ones we handed out), as long as it is correct you'll get full credit. Of course what constitutes correct is hard to specify in general but once the grading is done, please take a look at the grading rubric, which will be much more specific about what will get you a level 5.

      As another note, while our solutions are formatted and broken up using lemmas etc., your solution does **not** need to do so. As long as your solution *precisely* argues what it needed to (either with formal mathematical notation OR in English), with each step in your proof justified, then you'll receive full credit.

Please feel free to use the comment section to ask any followup question(s)!

homework1 homework2

Edit · good note | 0 · Updated 2 minutes ago by Atri Rudra

Story behind HW 1 Q2

note @152 stop following 3 views

Actions

Story behind the HW #1: Q2 on HW 1

Throughout the course there will be HW problems based on some really cool algorithmic idea (at least according to me!) that has some real life application and/or is something that I have used in my research. After the solutions for the corresponding HW have been handed out, I'll followup with a post on piazza giving more pointers for the connection. This is the first one in the series and is related to Q2 on HW 1.

I have had Q2 on HW 1 for all the years I have taught CSE 331. Until summer 2018, the best known upper bound was around $O((n!)^{2/3})$ (source), which is way worse the the best known lower bound, which is of the form c^n for some constant $c > 1$ (in Q2 you showed $c = \sqrt{2}$).

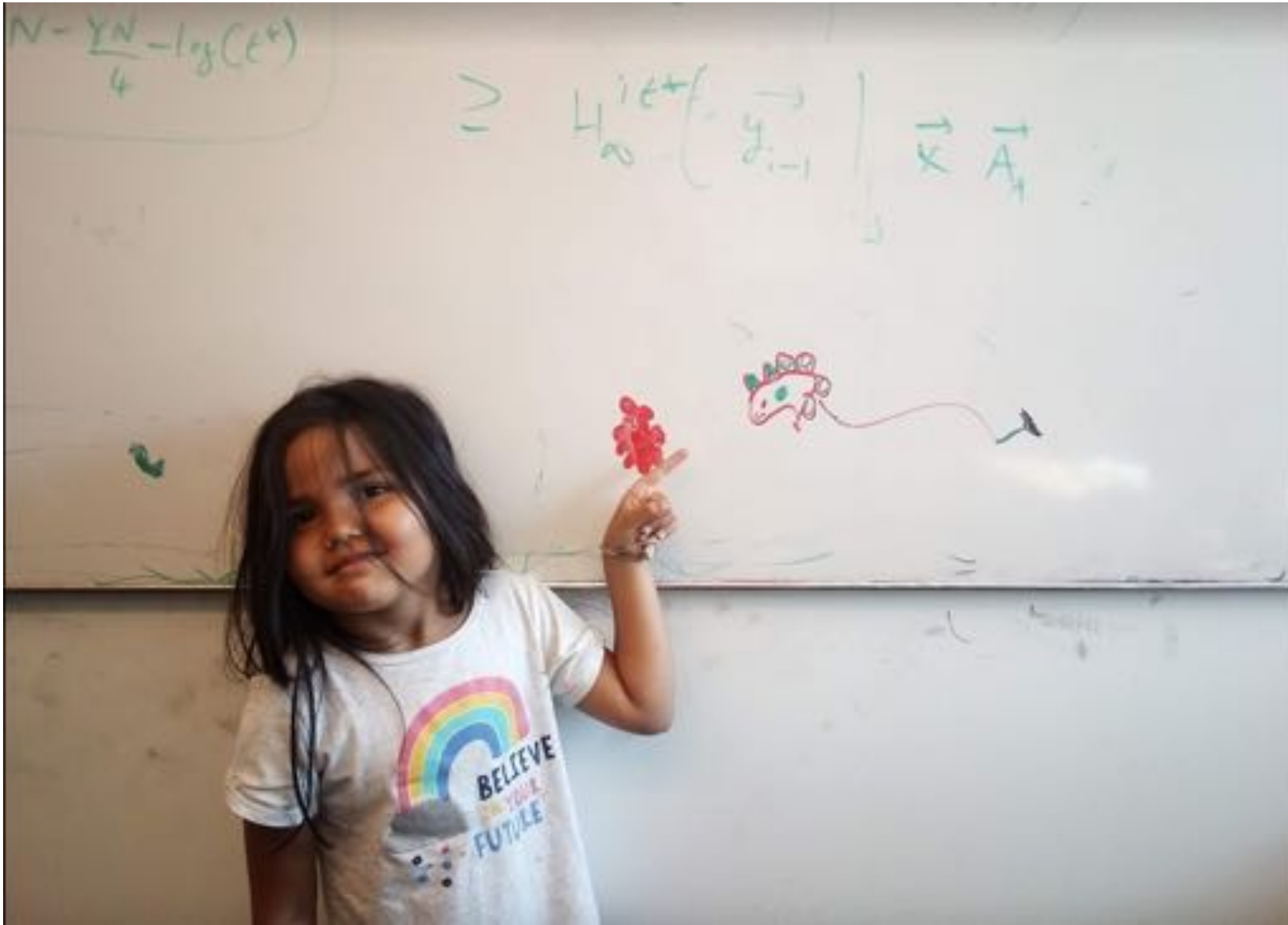
Over summer 18, a paper was presented which showed that the upper bound was C^n for some constant C. There is still a gap but the game now is to figure out the correct base C.

storybehindhw homework1

Edit good note 0

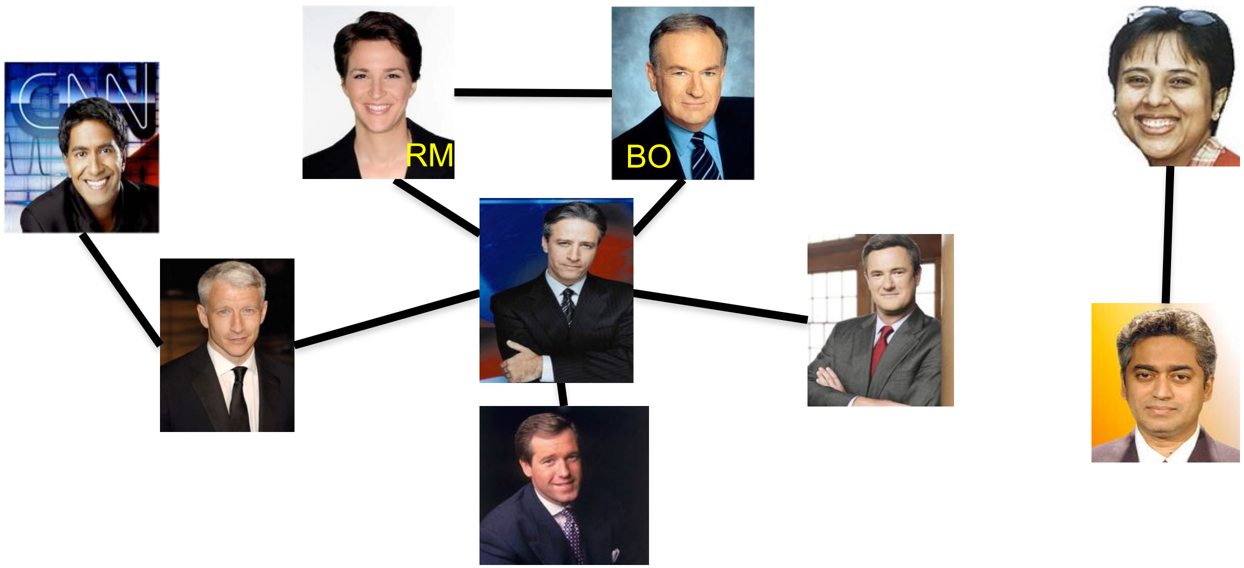
Updated 2 minutes ago by Atri Rudra

Questions/Comments?



Distance between u and v

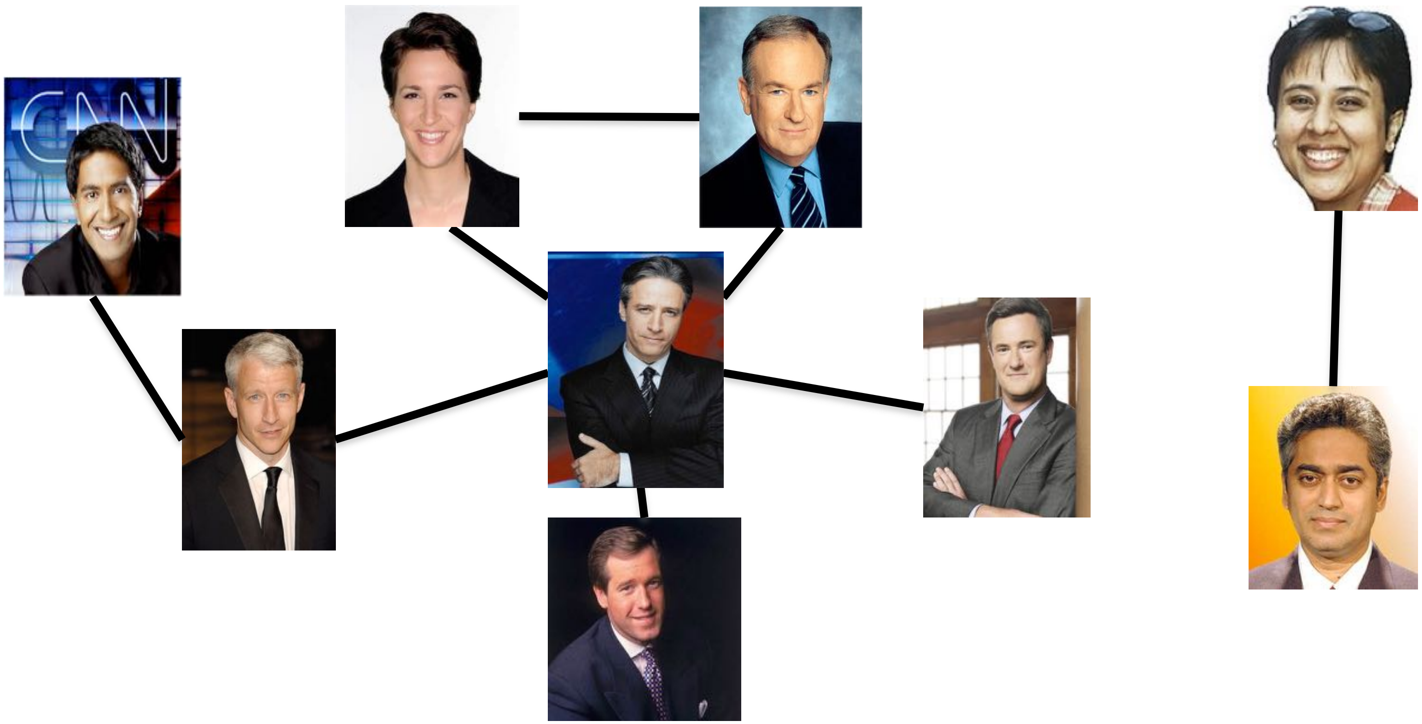
Length of the shortest length path between u and v



Distance between RM and BO? 1

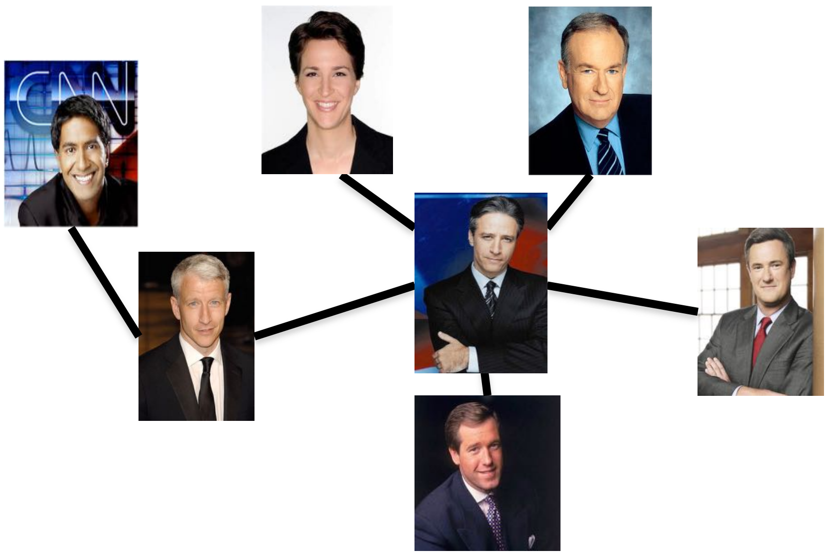
Tree

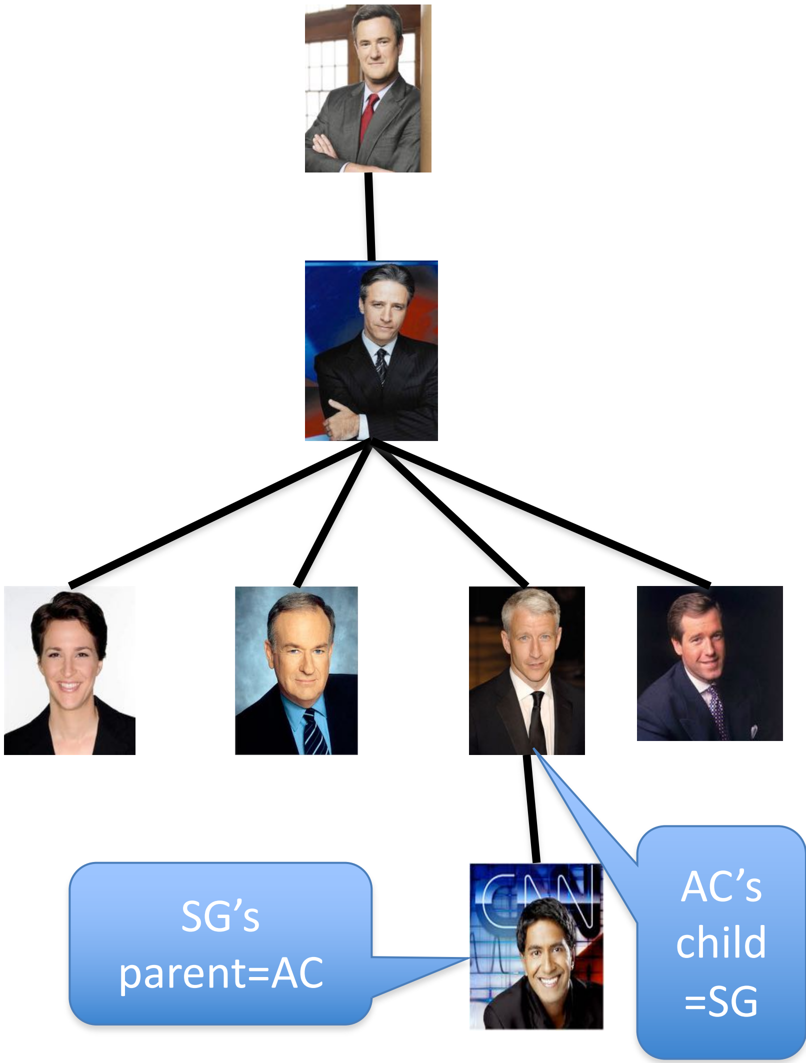
Connected undirected graph with no cycles



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

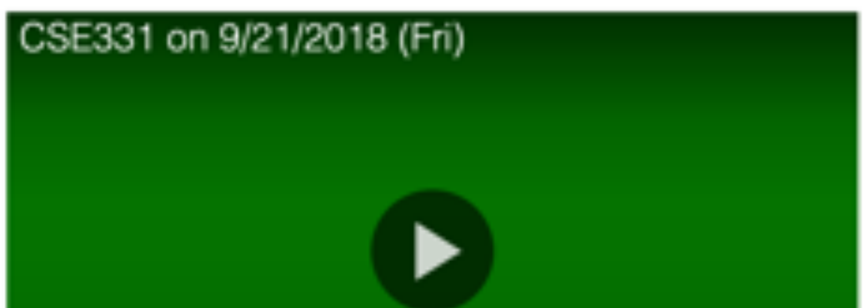
This page collects material from previous incarnations of CSE 331 on trees, especially the proof that trees with n nodes have exactly $n-1$ edges.

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 T be an undirected graph on n nodes

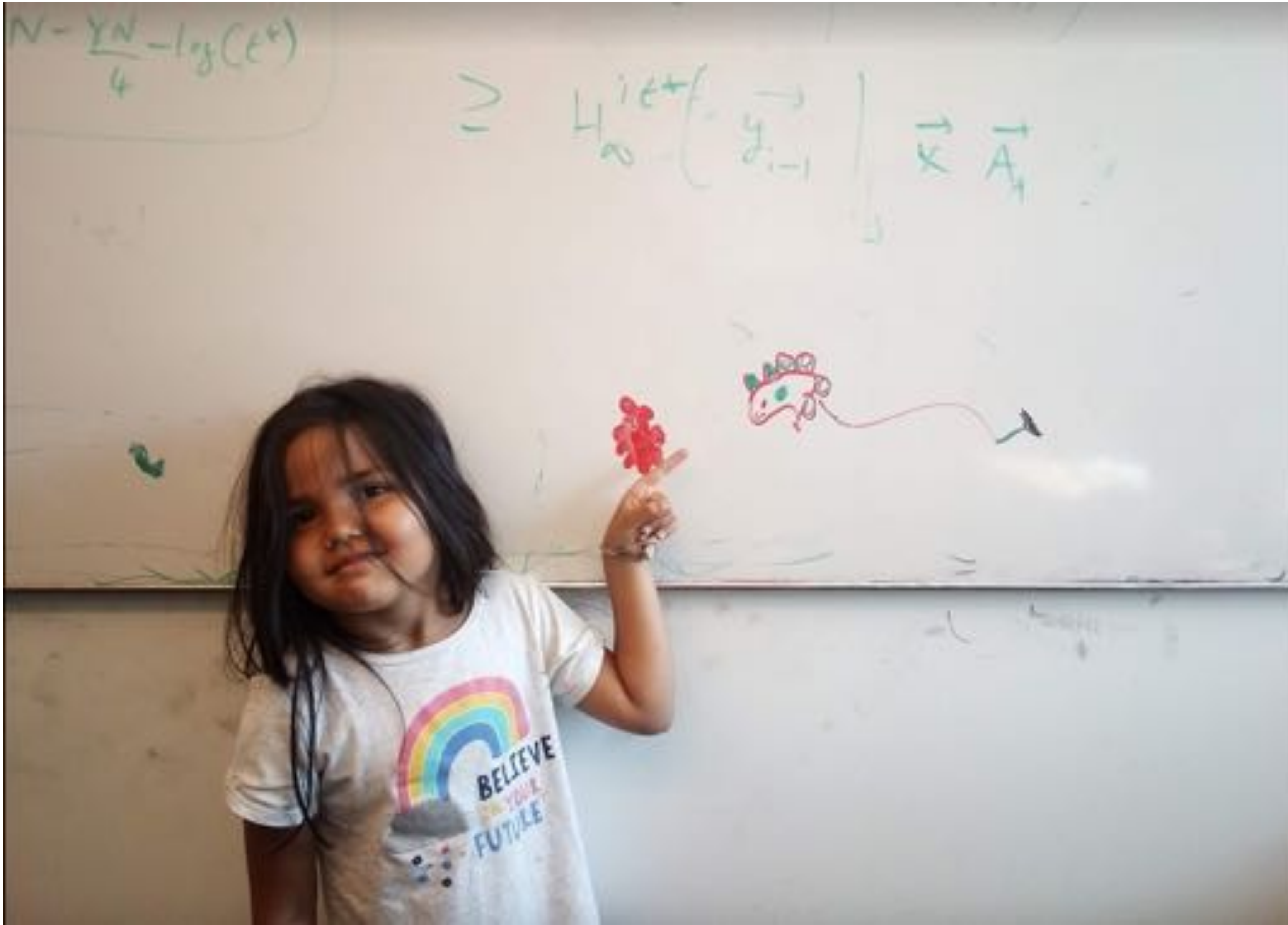
Then ANY two of the following implies the third:

T is connected

T has no cycles

T has n-1 edges

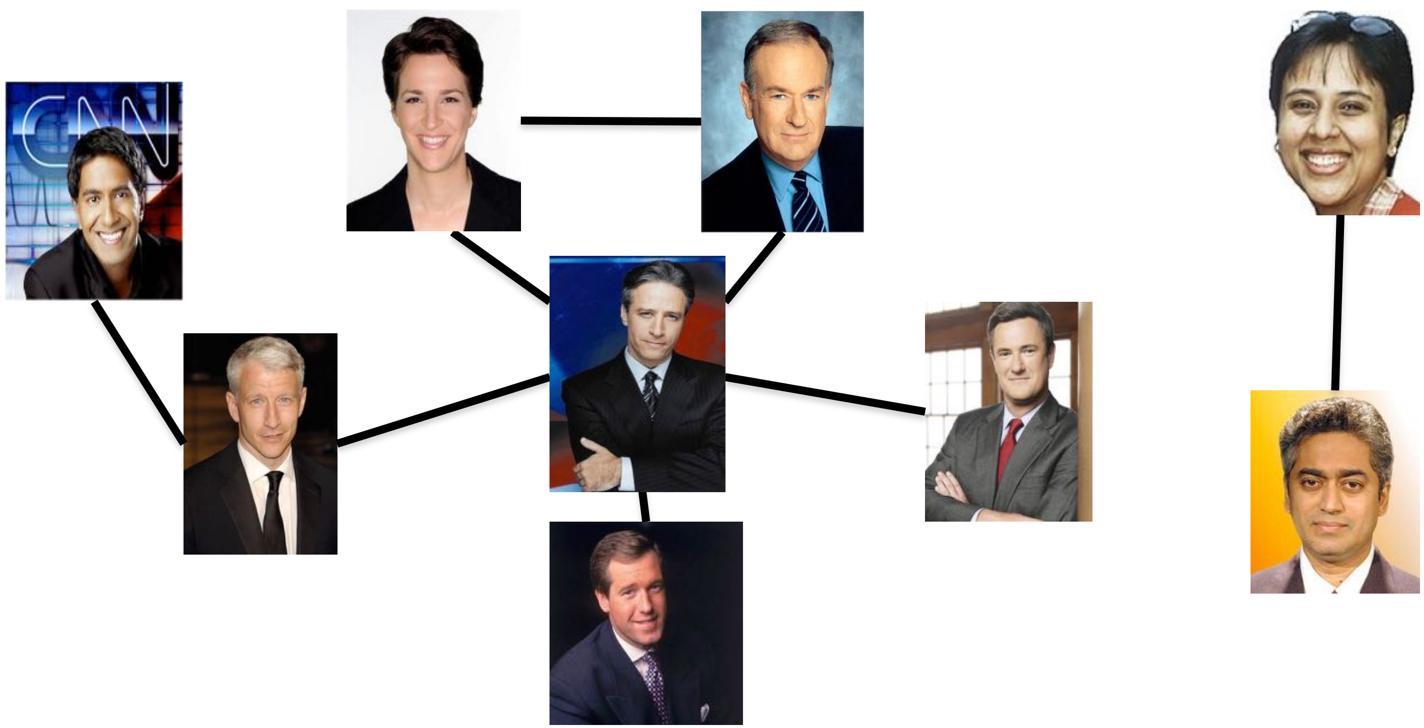
Questions/Comments?



Rest of Today's agenda

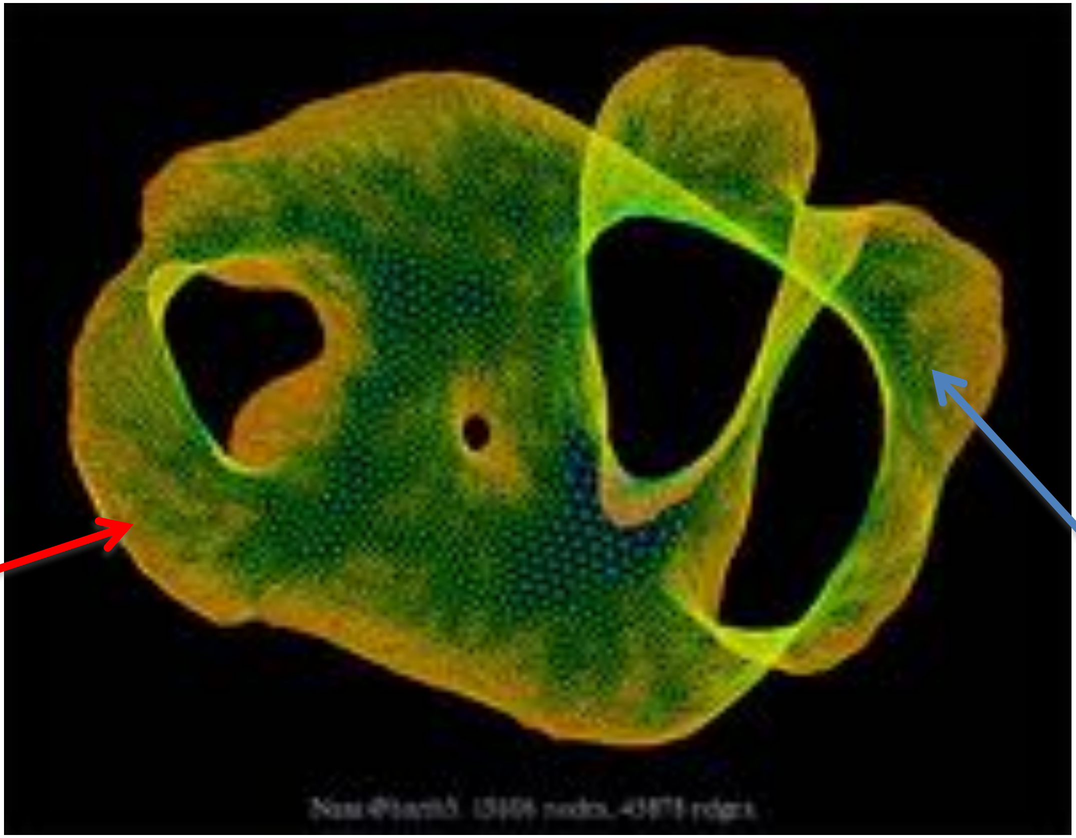
Algorithms for checking connectivity

Checking by inspection



What about large graphs?

s



t

Are s and t connected?

Brute-force algorithm?

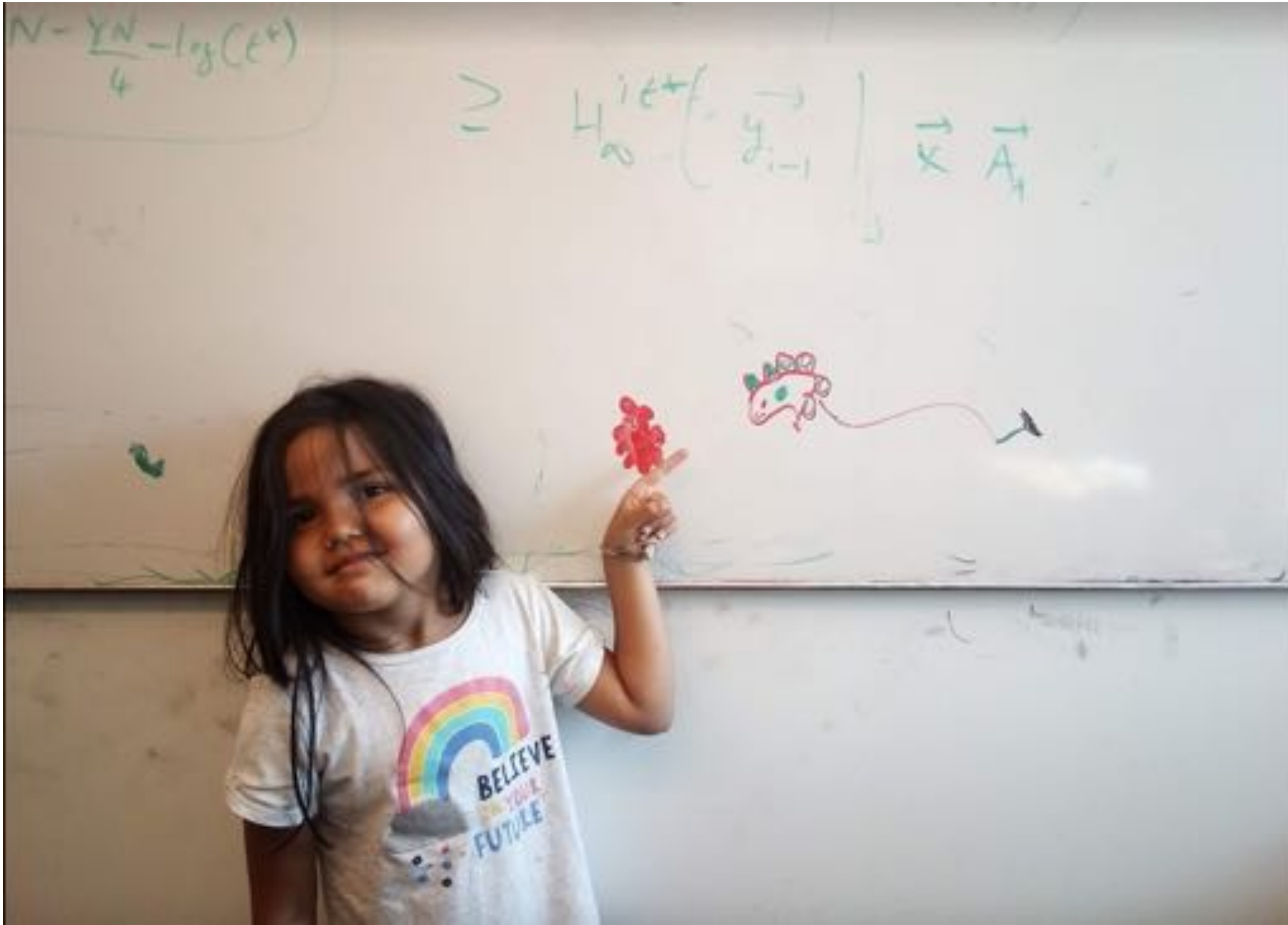
List all possible vertex sequences between s and t

n^n such sequences

Check if any is a path between s and t

Algorithm motivation

Questions/Comments?



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

Connected component of s

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

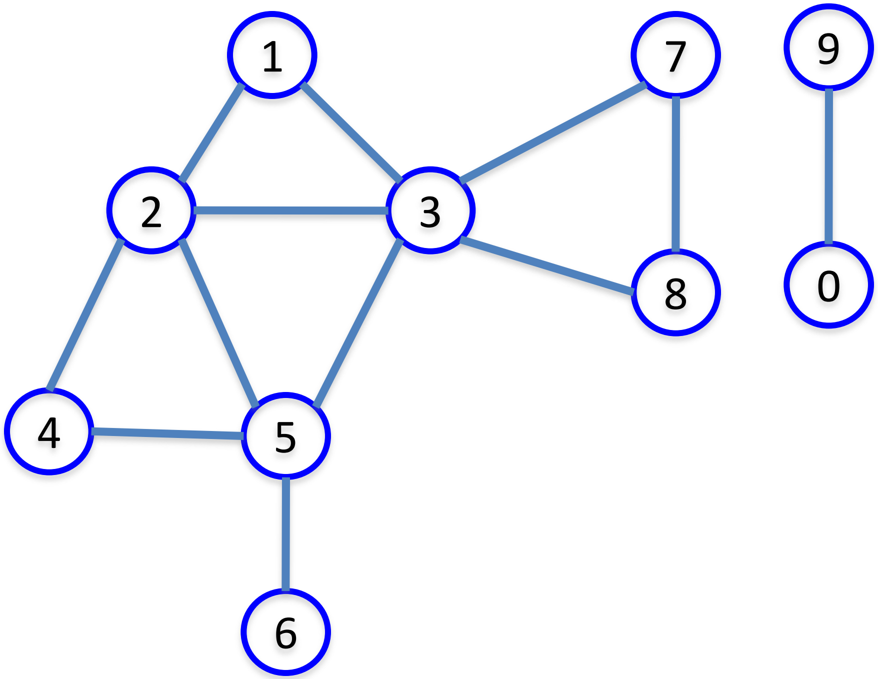
Stop when new layer is empty

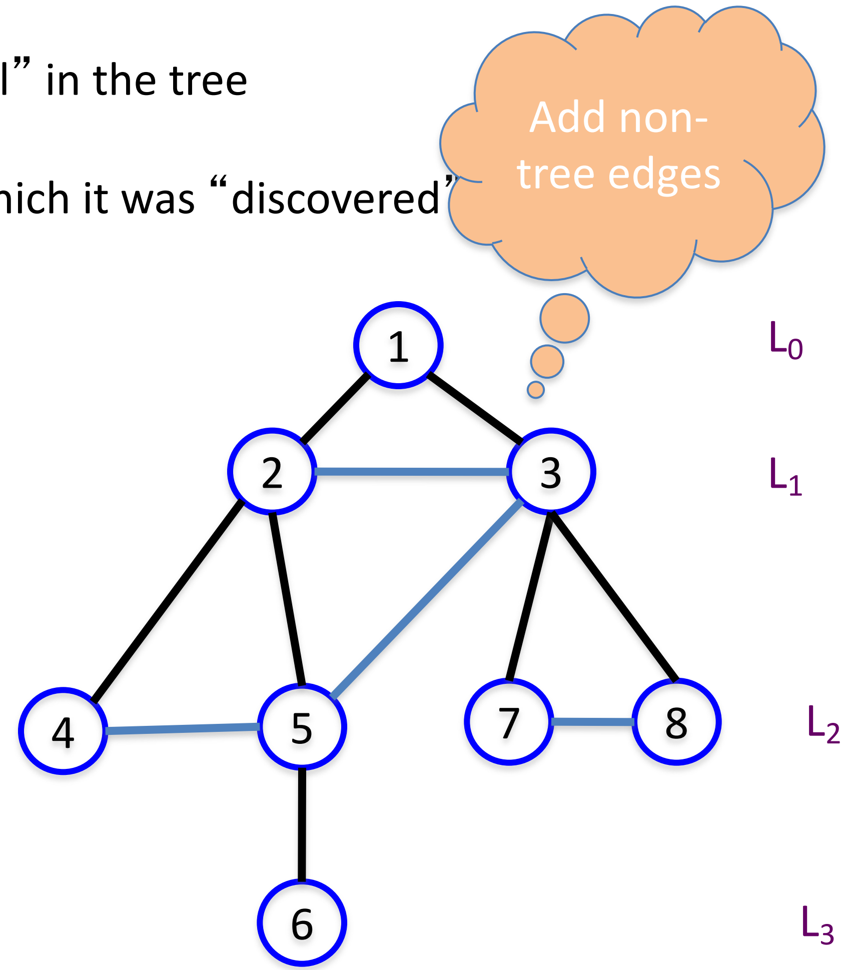
BFS Tree

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"





Argue on the board…