

# Lecture 12

CSE 331

Sep 26, 2022

If you need it, ask for help



# Project groups due **FRIDAY!**

**Deadline: Friday, Sep 30, 11:59pm**

CSE 331

Syllabus

Piazza

Schedule

Homeworks -

Autolab

Project -

Support Pages -

channel

Sample Exams -

## Forming groups

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

Project Overview

Group signup form

• 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!

If you do not fill in the form for group composition by the deadline, then you get a zero for the entire project.

# Upcoming quiz/exams

Quiz 1 Friday NEXT week

Mid-term 1 Monday in TWO weeks

Mid-term 2 Wed two days after Mid-term 1

Piazza post (+sample mid-terms) up by Thur. on preparing for mid-terms

# Clarifications on Q3

note @169 stop following 3 views Actions

## Regarding Q3

Some reminder/clarifications on Q3 on the HWs:

- See the [autolab page](#) section "Dealing with Errors" on how to handle two of the more common errors students encounter when submitting code for Q3 on Autolab.
  - Also see [@121](#)
  - Note that if you want Autolab to only run your code until say input  $x$  for some integer  $x < 10$  (e.g. to get partial credit), set `MaxInputs = x`
- The actual testcases were created more than 5 years ago so at this point I have completely forgotten how they were created so even if I wanted to I will not be able to give you all hints on what a specific testcase on a specific Q3 "looks like". All the feedback we can give on Q3 is what you get from Autolab.
  - Note that for each Q3 we will tell you the target runtime complexity.
- As I have mentioned few times, please do not spend a lot of time on Q3 since they can be a time sink. In particular, I strongly recommend that you do not look at Q3 until you are done with Q1 and at least Q2(a).

[homework2](#)

[Edit](#) good note 0

Updated 1 minute ago by Atri Rudra

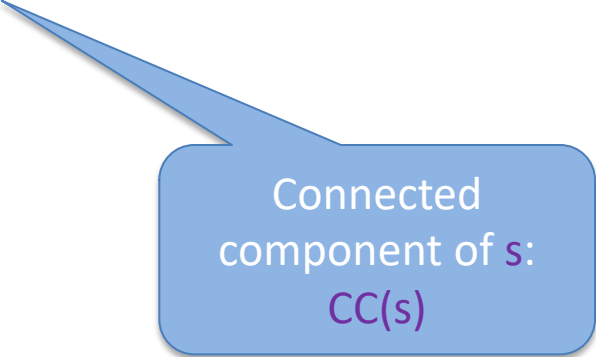
# Questions?



# Connectivity Problem

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

*Output:* All  $t$  connected to  $s$  in  $G$



Connected  
component of  $s$ :  
 $CC(s)$

# Breadth First Search (BFS)

Build layers of vertices connected to  $s$

$$L_0 = \{s\}$$

Assume  $L_0, \dots, L_j$  have been constructed

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

Stop when new layer is empty

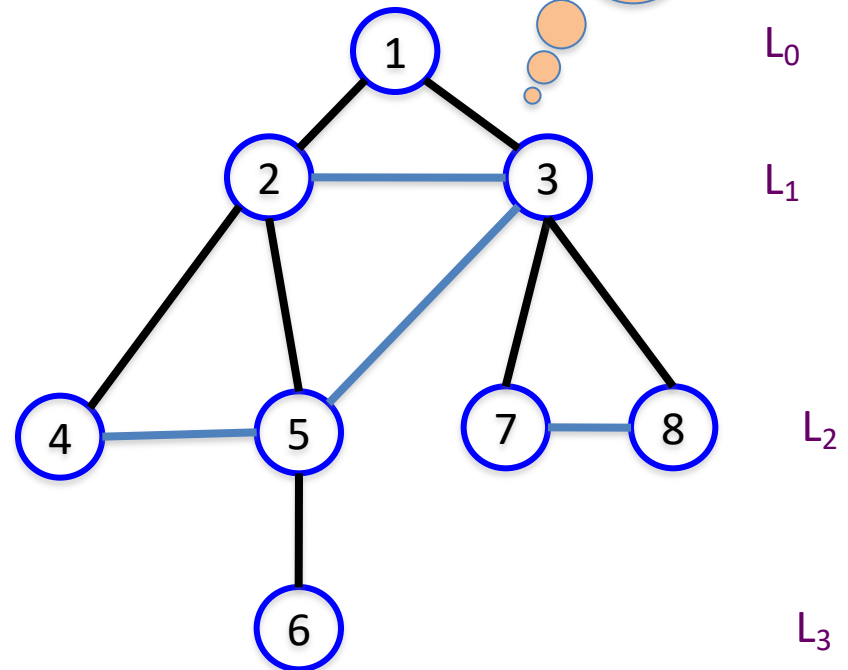
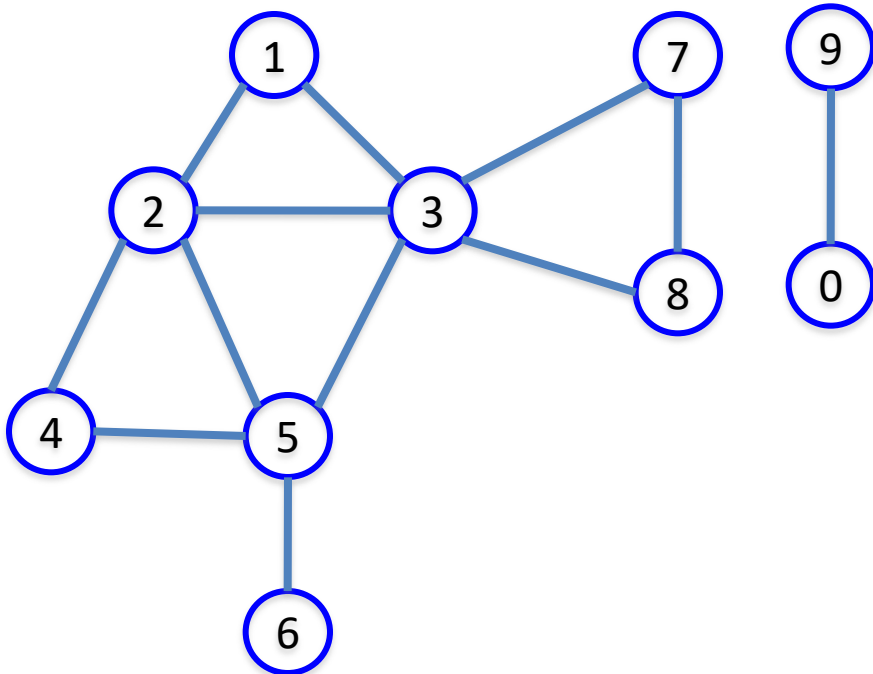


# BFS Tree

BFS naturally defines a tree rooted at  $s$

$L_j$  forms the  $j$ th “level” in the tree

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



# Two facts about BFS trees

All non-tree edges are in the same or consecutive layer

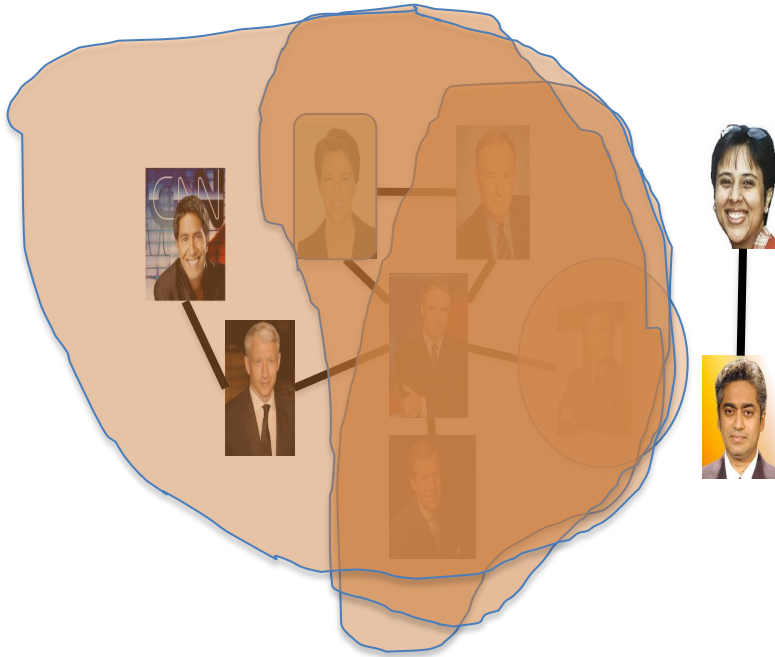
If  $u$  is in  $L_i$  then  $\text{dist}(s,u) = i$



# Rest of today's agenda

Computing Connected component

# Computing Connected Component



Explore( $s$ )

Start with  $R = \{s\}$

While exists  $(u,w)$  edge  $w$  not in  $R$  and  $u$  in  $R$

Add  $w$  to  $R$

Output  $R^* = R$

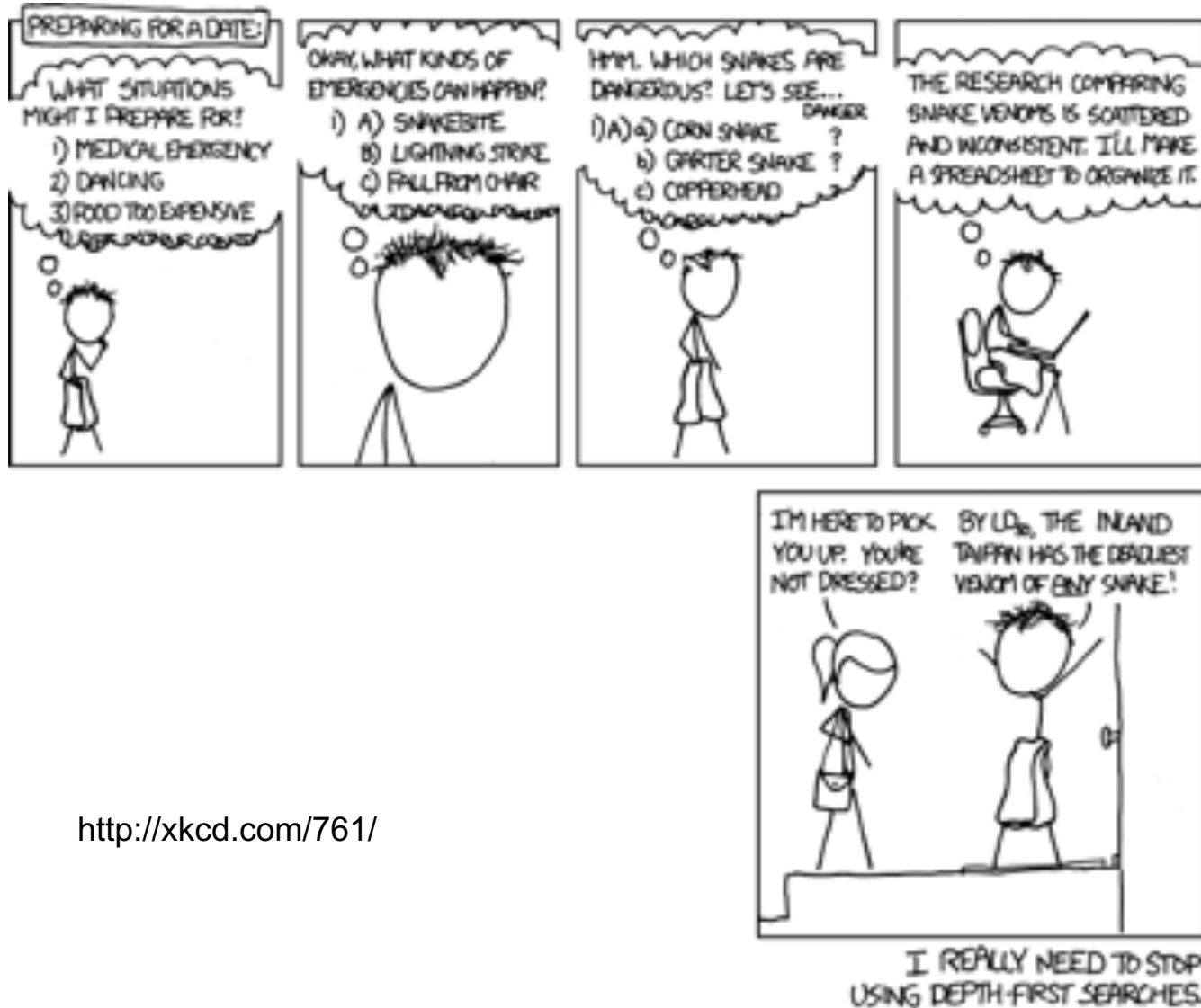
# Argue correctness on the board...



# BFS



# Depth First Search (DFS)



<http://xkcd.com/761/>



# DFS(**u**)

Mark **u** as explored and add **u** to **R**

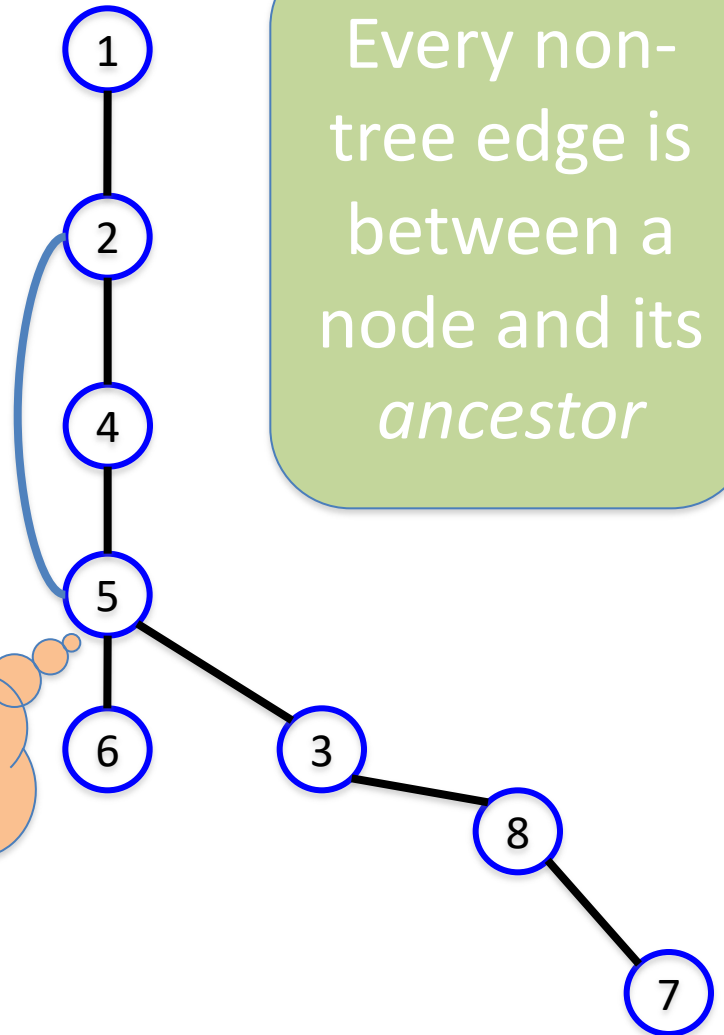
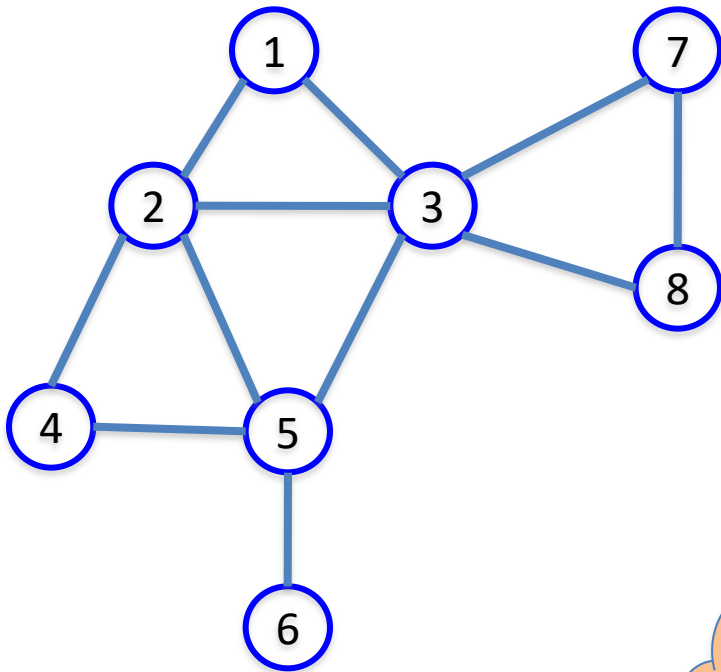
For each edge (**u**,**v**)

    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*

