Lecture 13

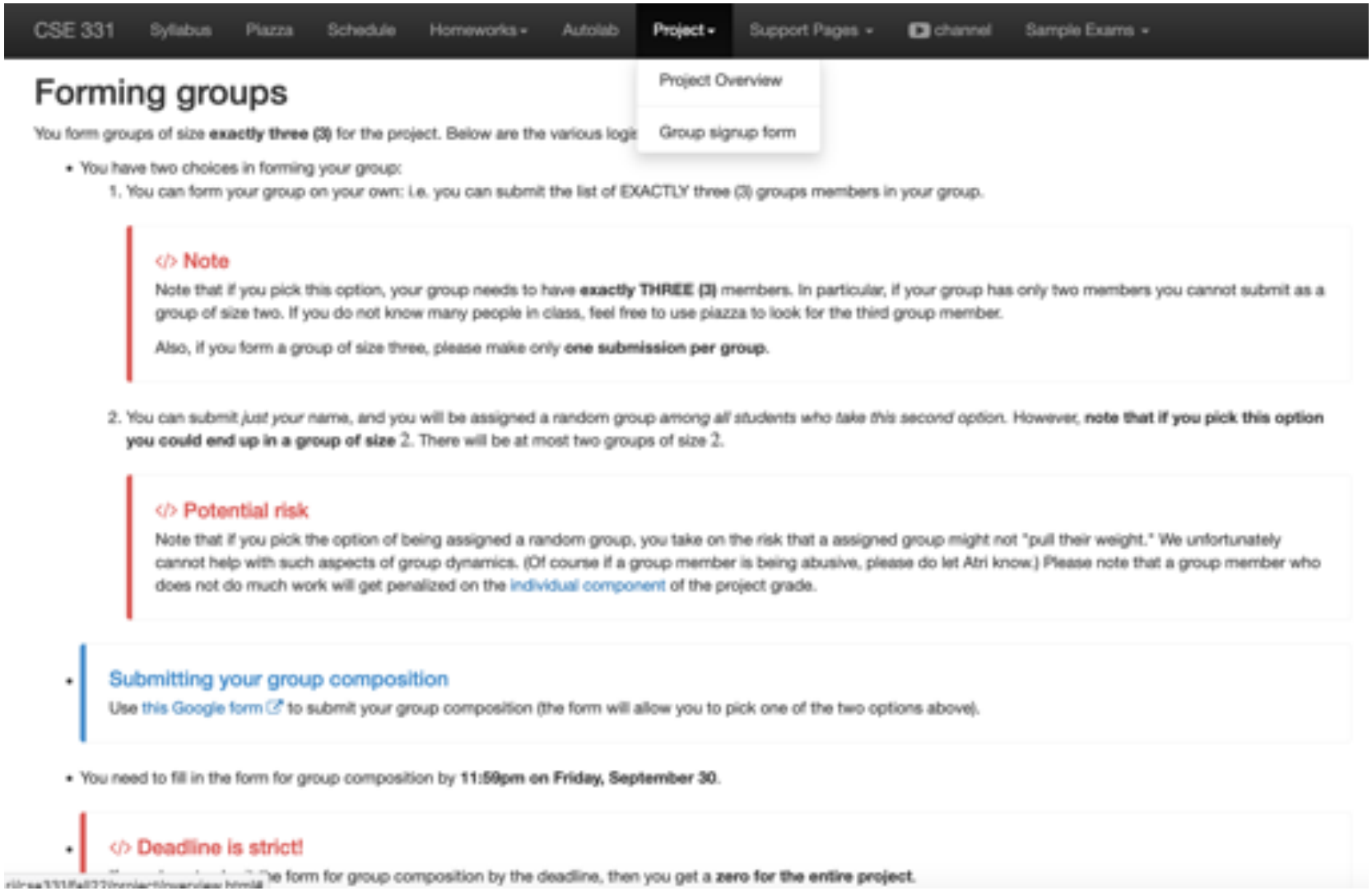
CSE 331

Sep 28, 2022

If you need it, ask for help

Project groups due FRIDAY!

Deadline: Friday, Sep 30, 11:59pm



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 up by tomorrow on preparing for mid-terms

Sample mid-terms

note @182 stop following 2 views

Actions

Sample mid-term exams

You can access the two sample mid-terms (and their solutions) from the navbar on the CSE 331 webpage:

http://www-student.cse.buffalo.edu/~atri/cse331/fall22/index.html

I would highly recommend that you do **not** peek into the solutions till you have tried to solve the sample mid-terms.

Here are the direct links:

- Sample mid-term 1 (and its solutions)
- Sample mid-term 2 (and its solutions)

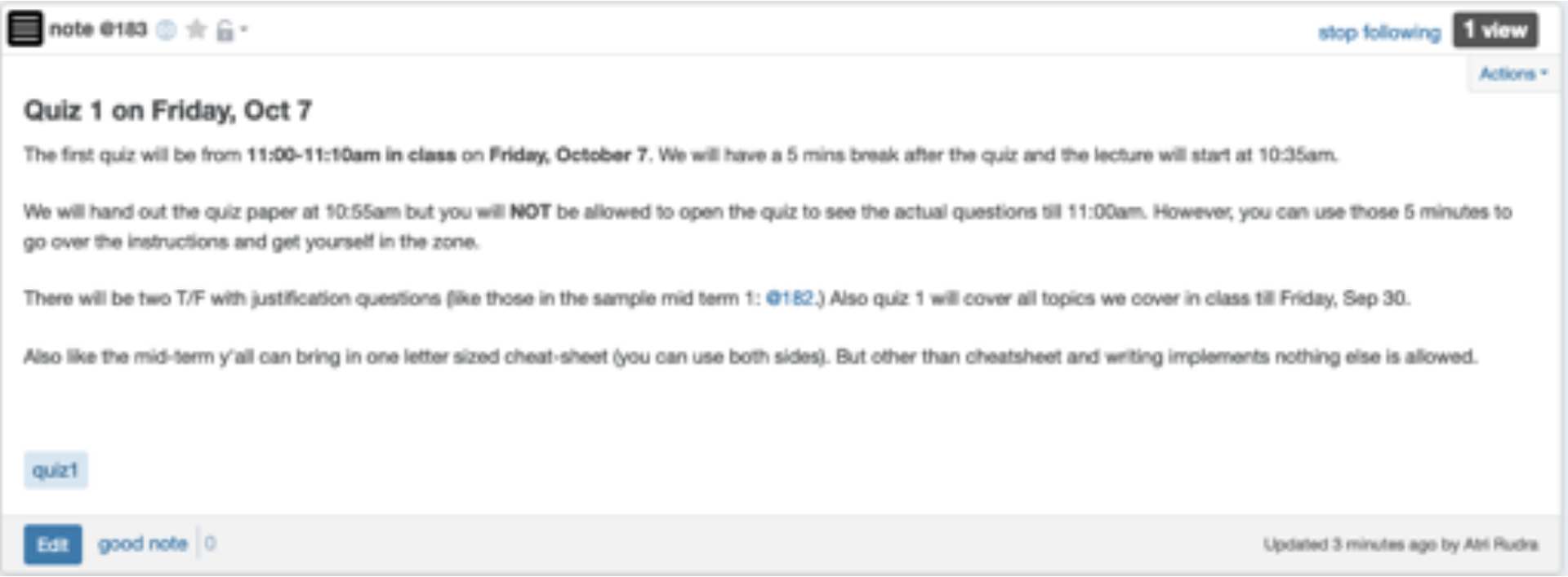
Over the next couple of days, I will put up a post on the mid-terms in general (what topics will be on it, some thoughts on how to prepare and so on). But for now, here is one important clarification: **The sample mid-terms are ONLY meant for y'all to get familiar with the format of the exams.** *You should not deduce ANYTHING about the coverage of topics or hardness of the exams from the sample mid-terms above.* (The mid-term post will talk more about the topics and the actual mid-terms will be harder than the sample exams above).

mid-term

Edit good note 0

Updated 4 minutes ago by Atri Rudra

Bit more on Quiz 1


note @183

stop following 1 view

Actions

Quiz 1 on Friday, Oct 7

The first quiz will be from **11:00-11:10am in class** on **Friday, October 7**. We will have a 5 mins break after the quiz and the lecture will start at 10:35am.

We will hand out the quiz paper at 10:55am but you will **NOT** be allowed to open the quiz to see the actual questions till 11:00am. However, you can use those 5 minutes to go over the instructions and get yourself in the zone.

There will be two T/F with justification questions (like those in the sample mid term 1: @182.) Also quiz 1 will cover all topics we cover in class till Friday, Sep 30.

Also like the mid-term y'all can bring in one letter sized cheat-sheet (you can use both sides). But other than cheatsheet and writing implements nothing else is allowed.

quiz1

Edit good note 0

Updated 3 minutes ago by Atri Rudra

HW 3 out

Homework 3

Due by **11:30pm, Tuesday, October 4, 2022**

Make sure you follow all the homework policies.

All submissions should be done via Autolab.

Sample Problem

The Problem

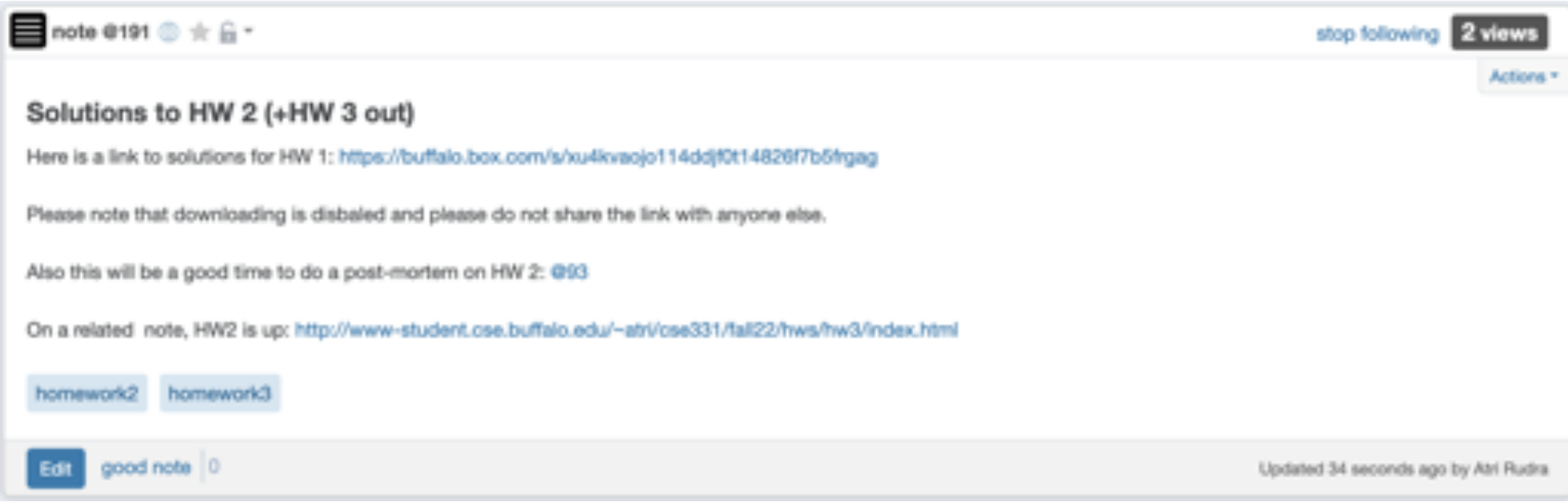
This problem is just to get you thinking about graphs and get more practice with proofs.

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).

! For those of you who are feeling a little ambitious

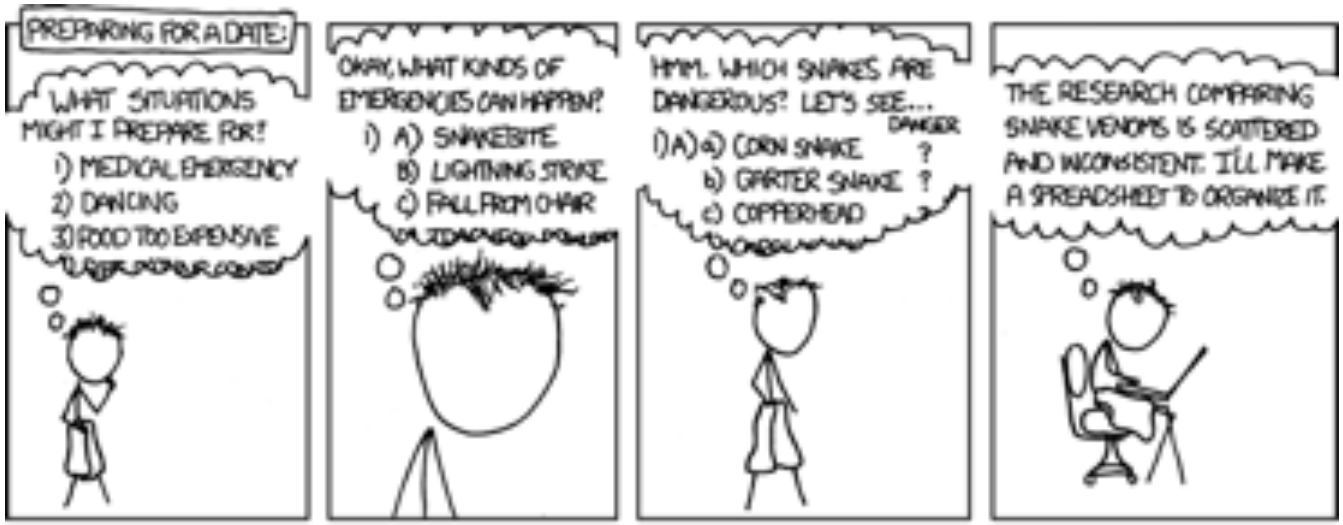
For the top 3 submissions in the scoreboard in Python, the top 2 submissions in the scoreboard in Java and the top submission in the scoreboard in C++, we are offering 2.5 bonus points. But be warned! You should not be spending too much time on this. We rather you work on Questions 1 and 2 above.

HW 2 solutions posted



Questions?

Depth First Search (DFS)



http://xkcd.com/761/

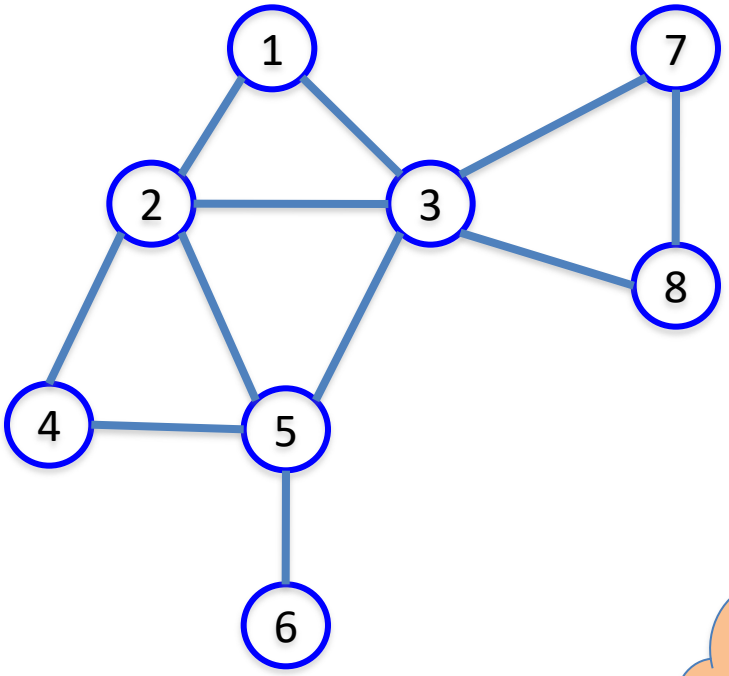
DFS(u)

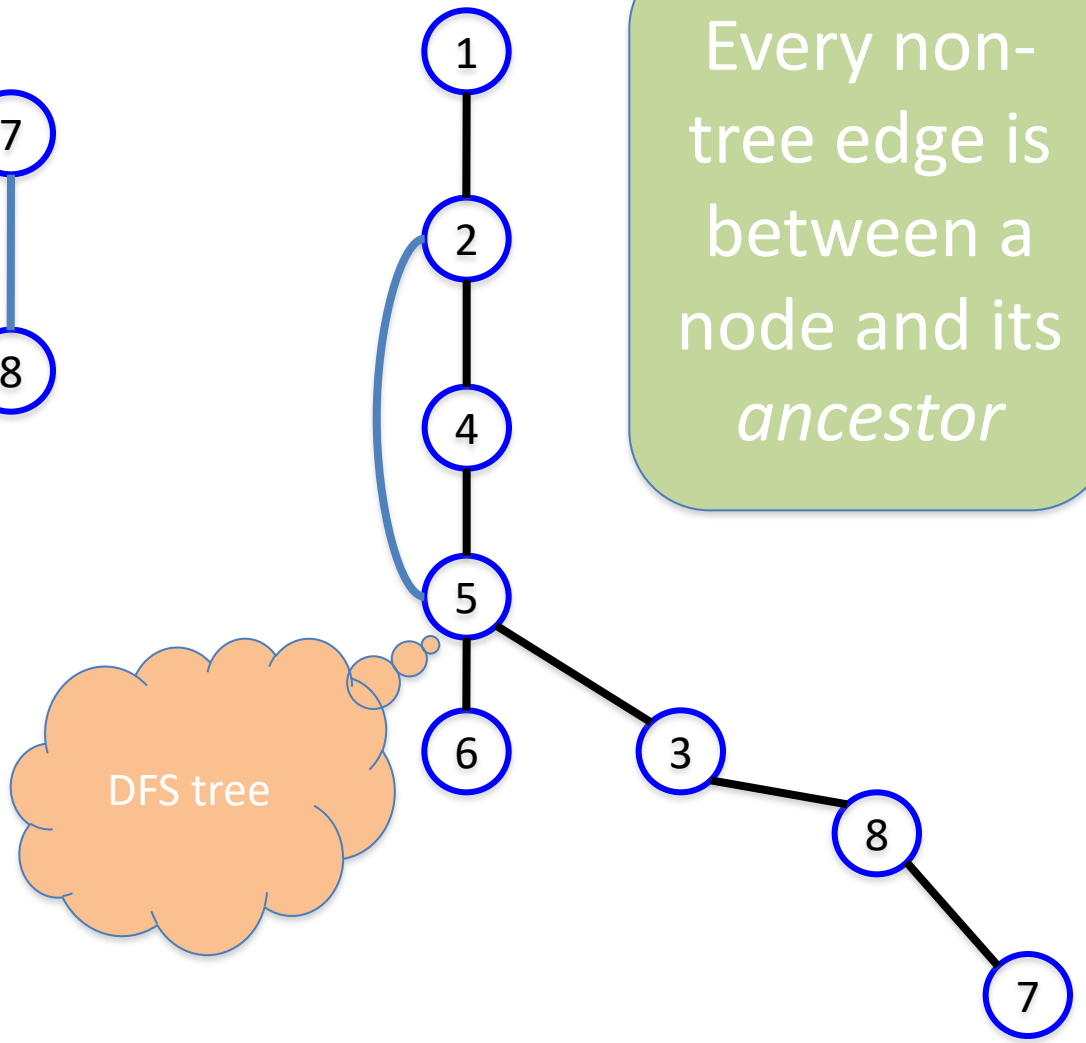
Mark u as explored and add u to R

For each edge (u,v)

If v is not explored then DFS(v)

A DFS run

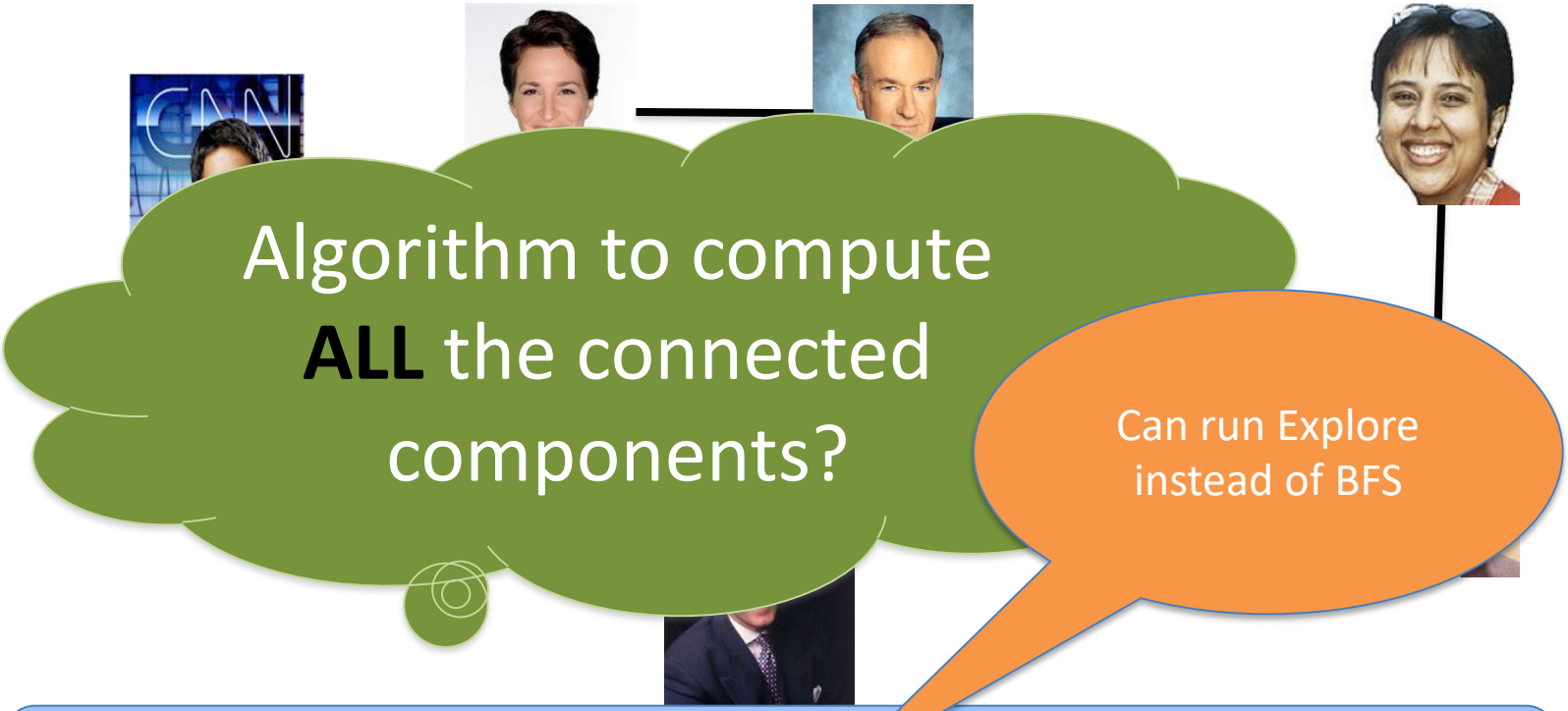




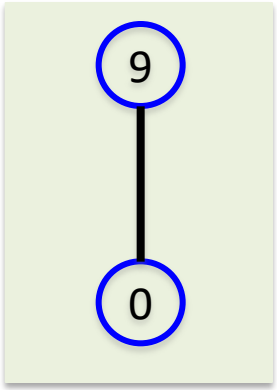
Every non-tree edge is between a node and its *ancestor*

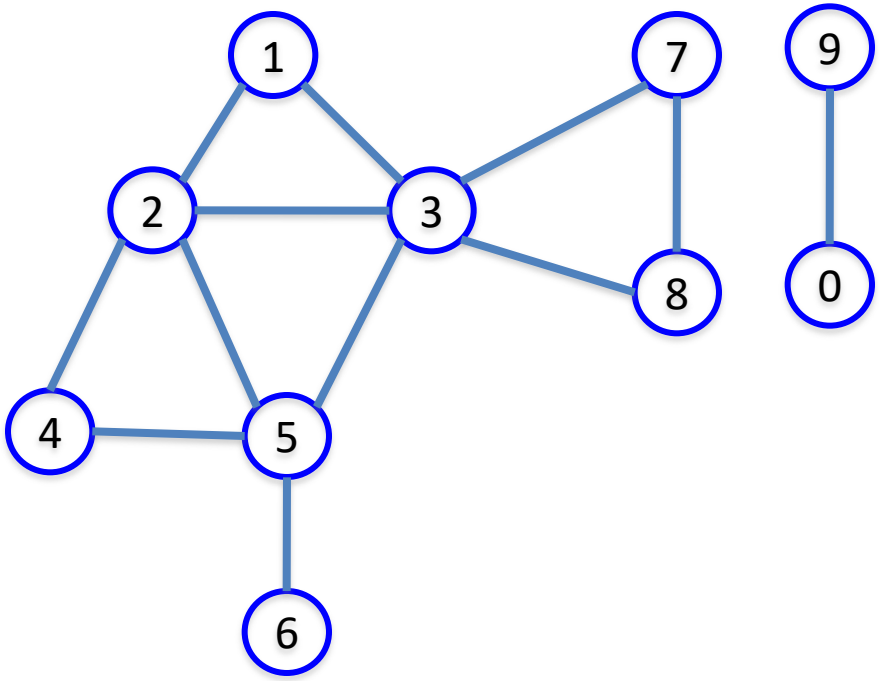
Connected components are disjoint

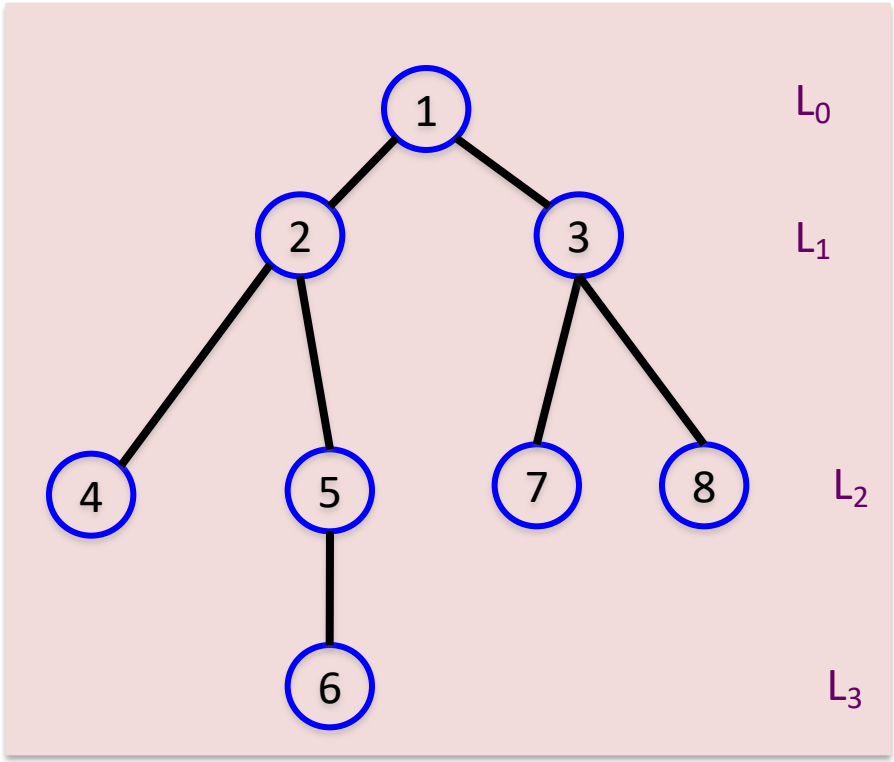
Either Connected components of s and t are the same or are disjoint



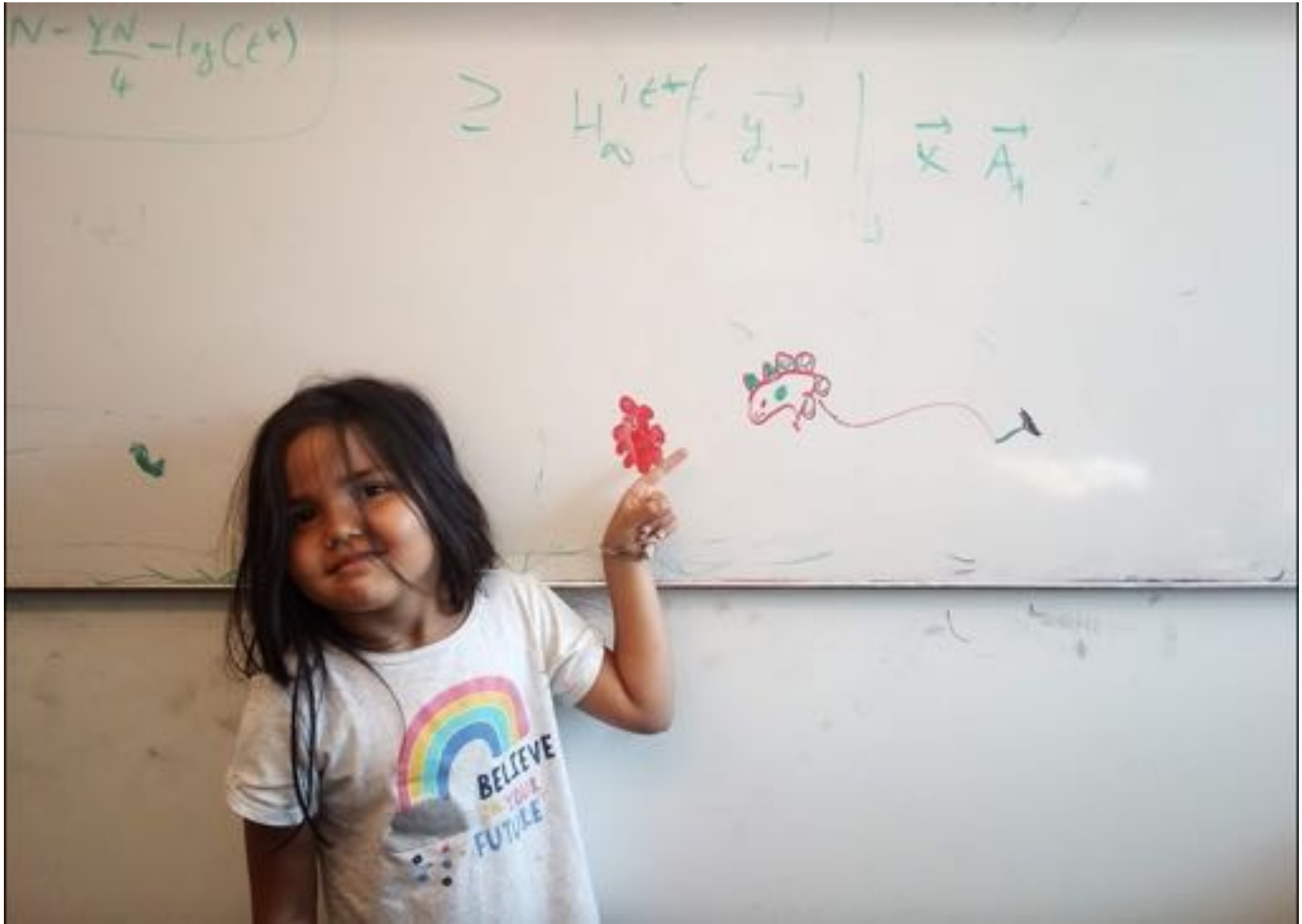
Computing all CCs







Questions/Comments?



Today's agenda

Run-time analysis of BFS (DFS)

Stacks and Queues

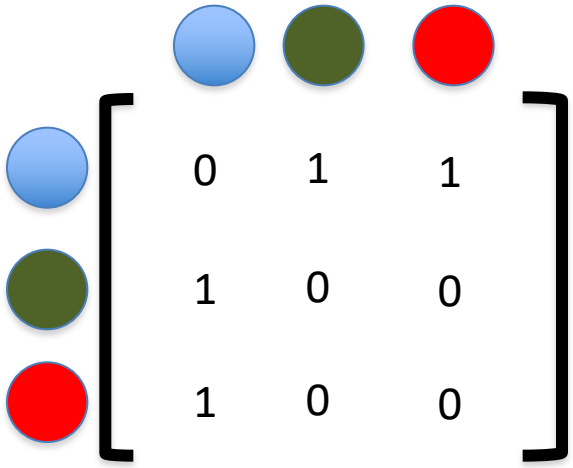
Last in First out

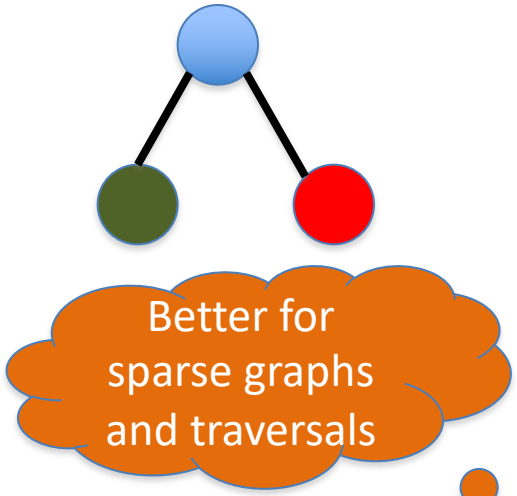
First in First out

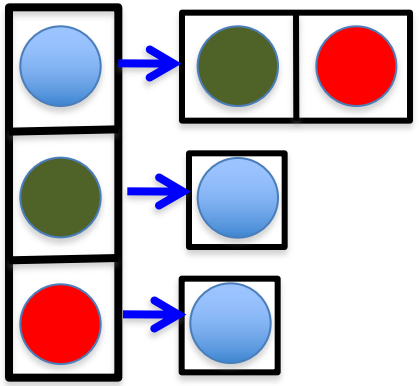
But first…

How do we represent graphs?

Graph representations

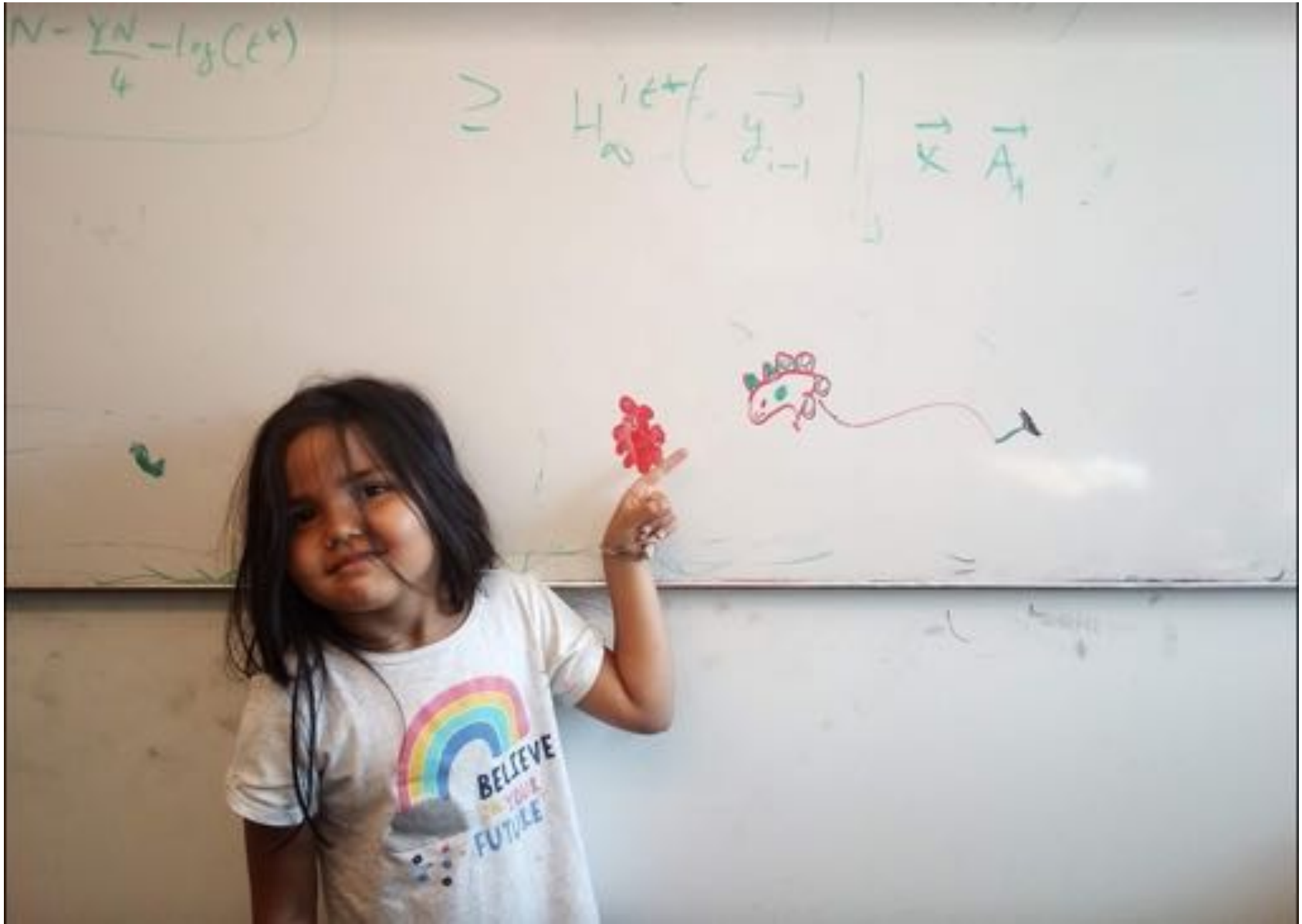






Adjacency matrix		Adjacency List
$O(1)$	(u,v) in E?	$O(n)$ [$O(n_v)$]
$O(n)$	All neighbors of u?	$O(n_u)$
$O(n^2)$	Space?	$O(m+n)$

Questions/Comments?

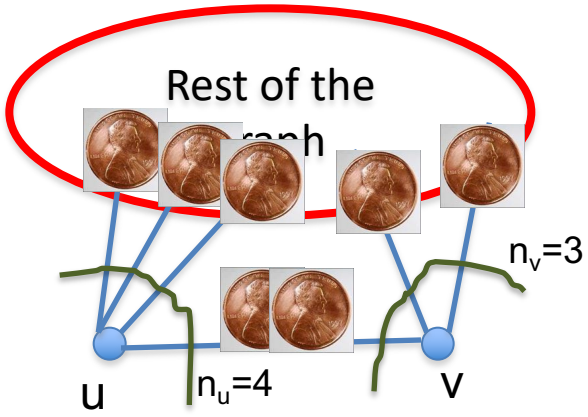


2·# edges = sum of # neighbors

$$2m = \Sigma_{u \text{ in } V} n_u$$

Give 2 pennies to each edge

Total # of pennies = 2m



Each edges gives one penny to its end points

# of pennies u receives = n_u

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

Use linked lists

Use CC[v] array

Rest of Today's agenda

Space complexity of Adjacency list representation

Quick run time analysis for BFS

Quick run time analysis for DFS (and Queue version of BFS)