Lecture 13

CSE 331

Please have a face mask on

Masking requirement

UB requires all students, employees and visitors – regardless of their vaccination status – to wear face coverings while inside campus buildings.

https://www.buffalo.edu/coronavirus/health-and-safety/health-safety-guidelines.html

Project groups due FRIDAY!

Deadline: Friday, March 4, 11:59pm

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

Project Overview

Group signup form

CSE 331 Project

Spring 2022

Details and motivations for the project.

Motivation

CSE 331 is primarily concerned with the technical aspects of algorithms: how to design them and then how to analyze their correctness and
in our world and is common place in many aspects of society. The main aim of the project is to have you explore in some depth some of the

Just to give some examples for such implications:

- Big data is hot these days and there is a (not uncommon) belief that by running (mainly machine learning) algorithms on big data, we
potentially make policy decisions. Here is a cautionary talk:

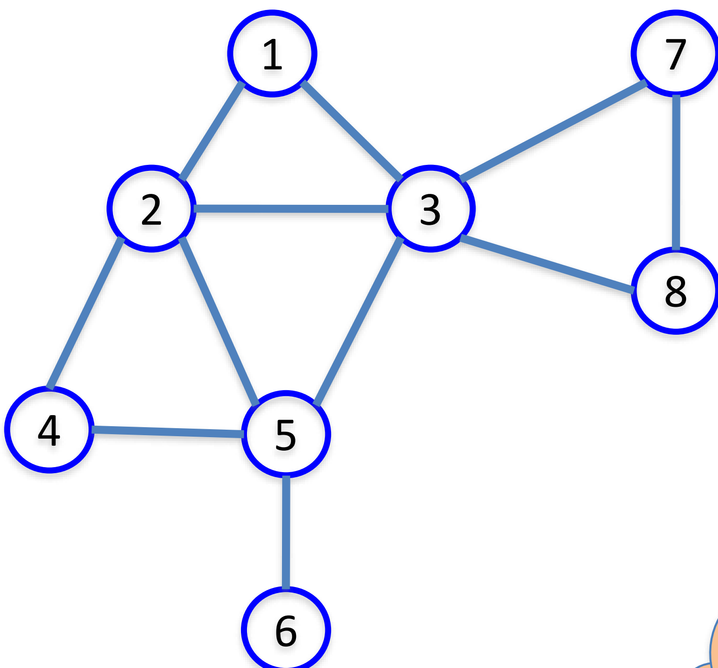
DFS(u)

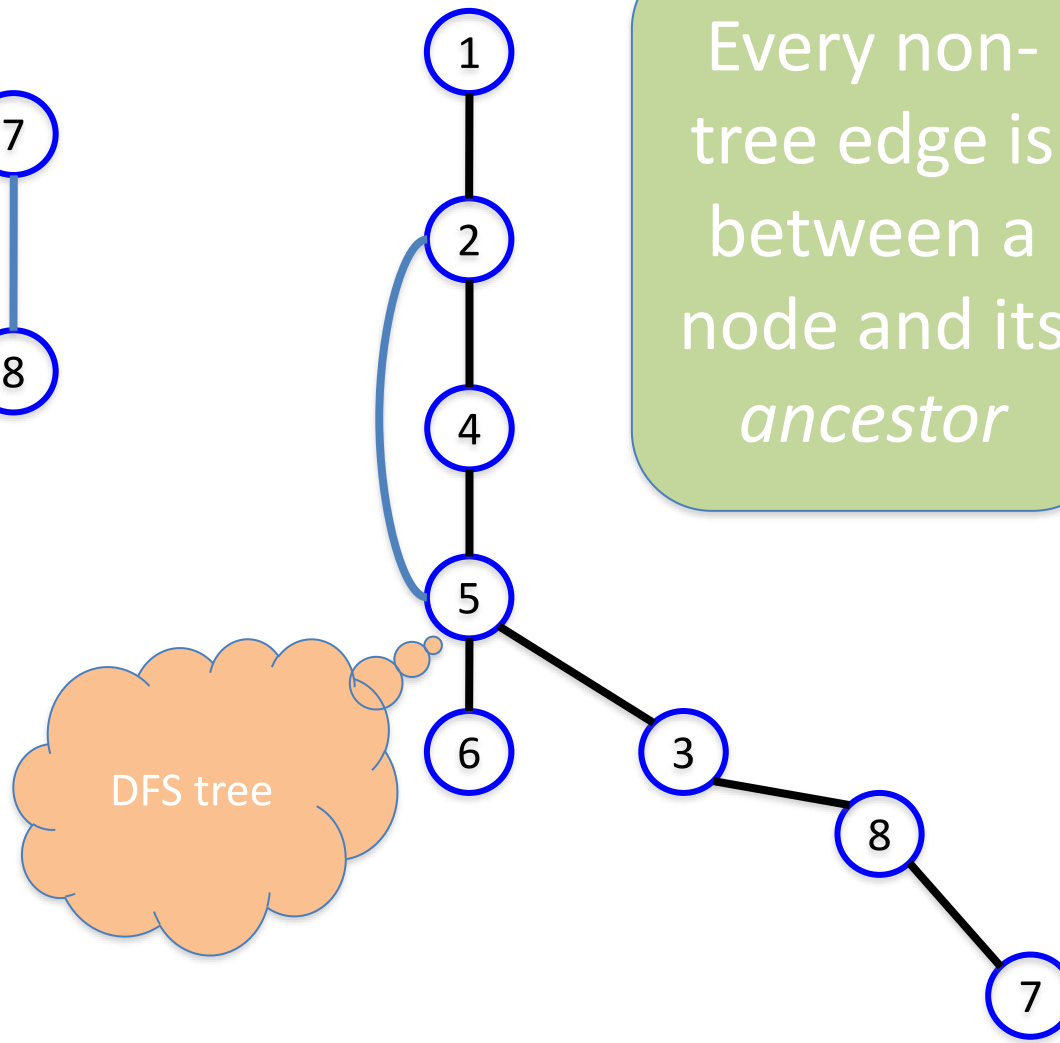
Mark u as explored and add u to R

For each edge (u,v)

If v is not explored then DFS(v)

A DFS run



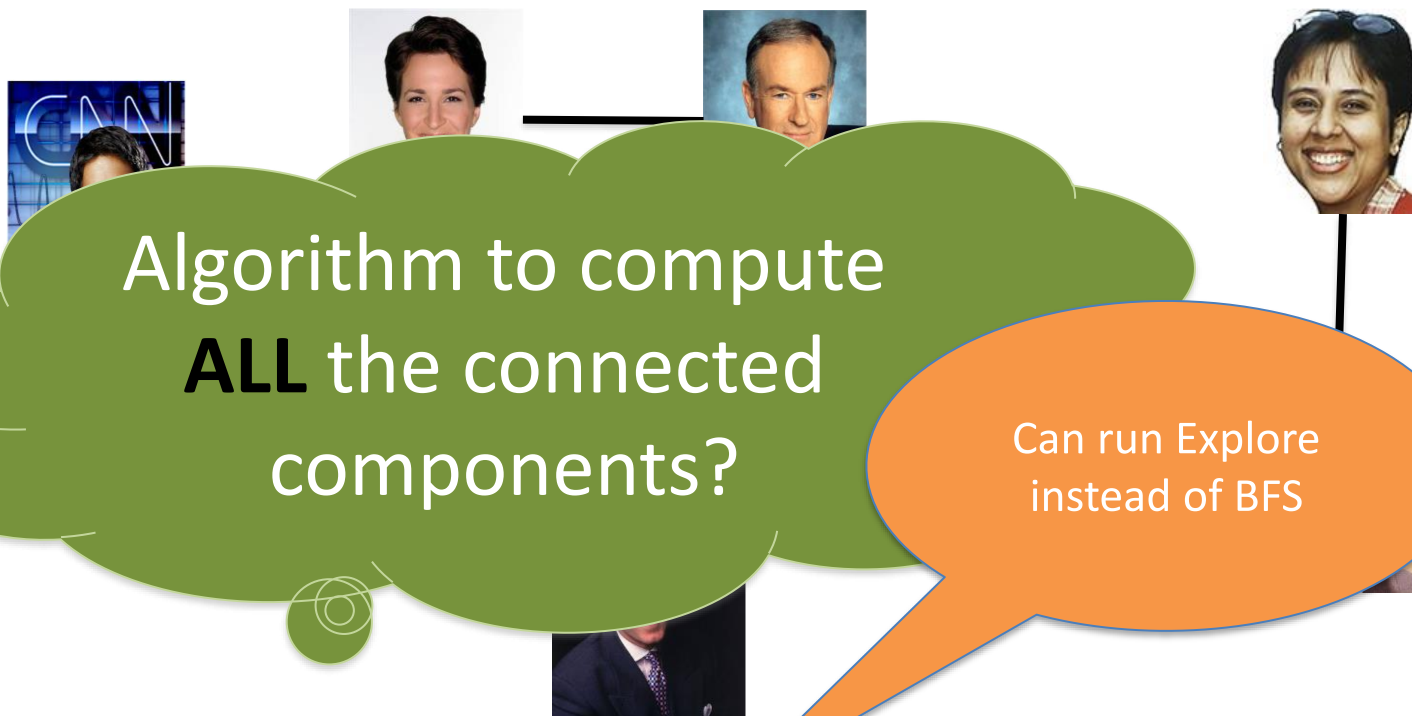


Every non-tree edge is between a node and its *ancestor*

DFS a special case of Explore

Connected components are disjoint

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

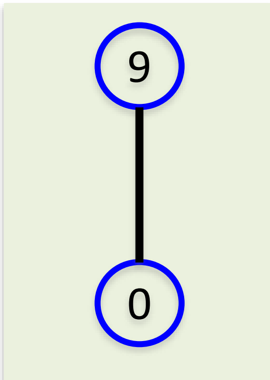


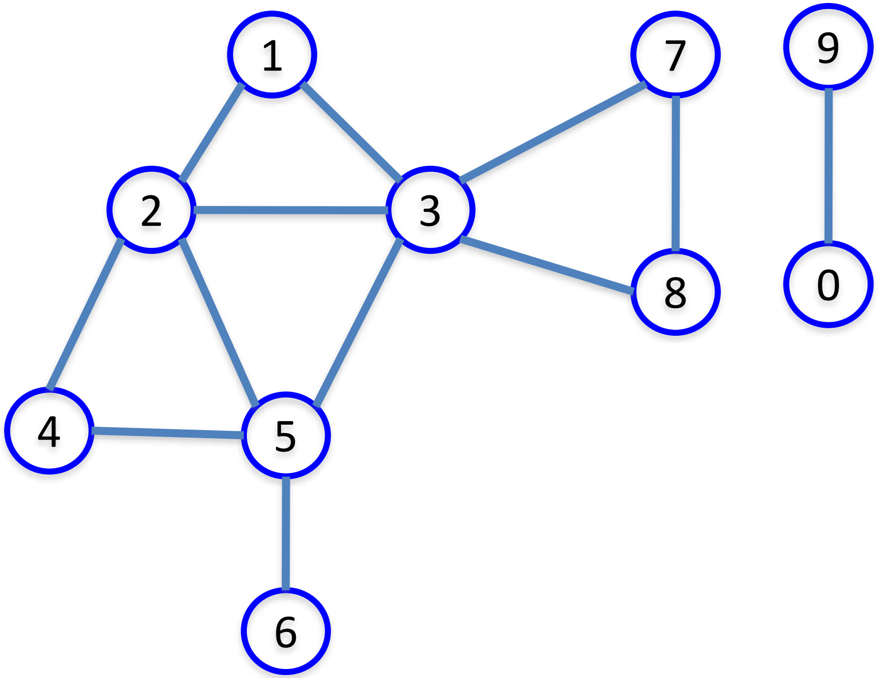
Algorithm to compute **ALL** the connected components?

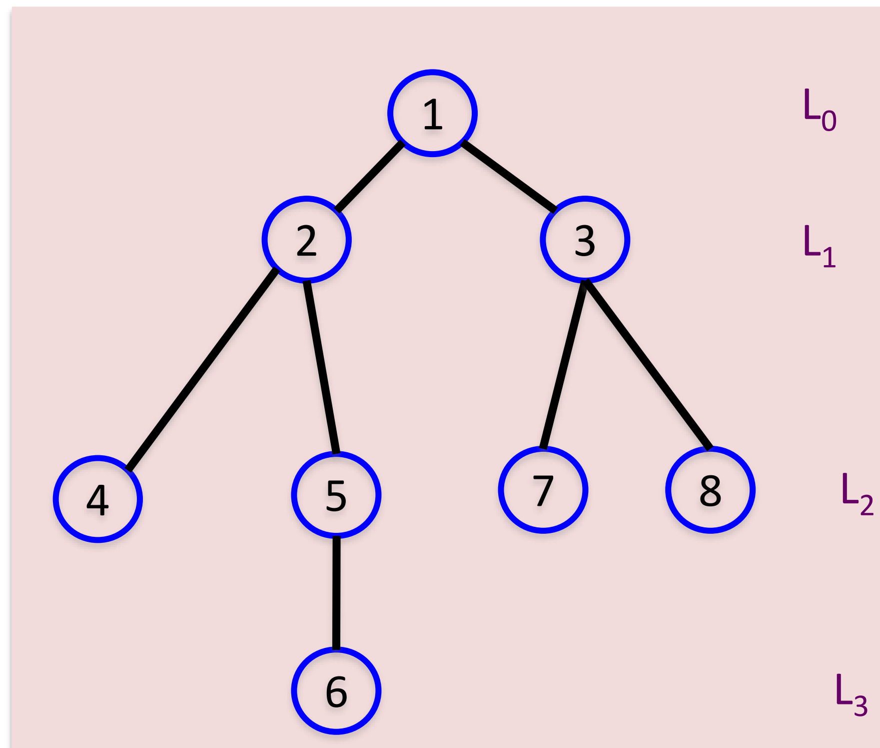
Can run Explore instead of BFS

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

Computing all CCs







Today's agenda

Run-time analysis of BFS (DFS)

Stacks and Queues

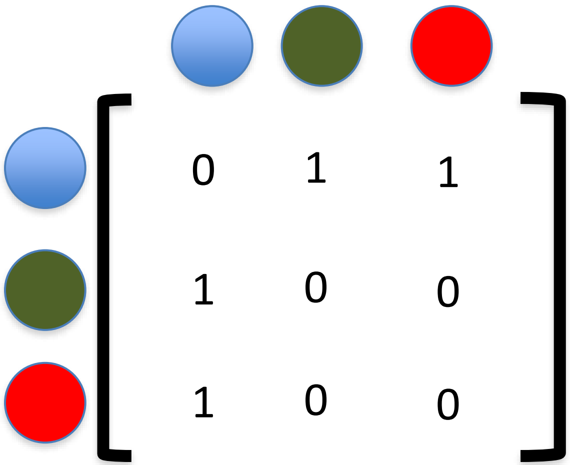
Last in First out

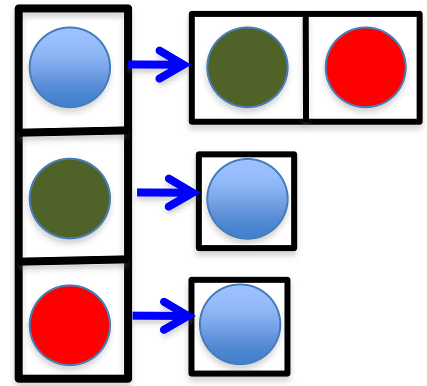
First in First out

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)

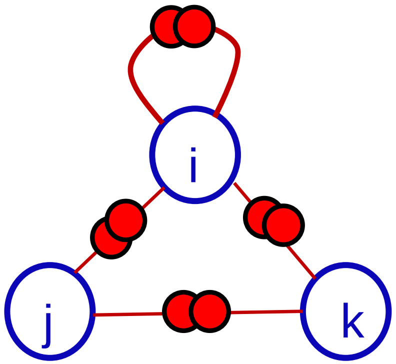
2·# edges = sum of # neighbors

2m = Σ u in V n_u

$$\Longrightarrow\ 2|E| = \sum_{u \in V} \deg(u)$$

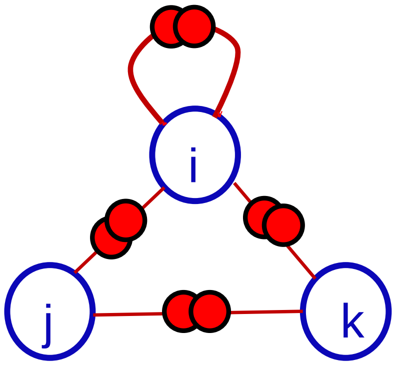
$$\sum_{u \in V} \deg(u) = \deg(u_1) + \deg(u_2) + \cdots + \deg(u_n)$$
$$= n_{u_1} + n_{u_2} + \ldots + n_{u_n}$$

Suppose we put two dots on every edge.

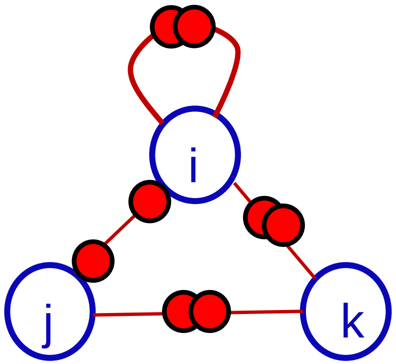


How many dots are there? 2 dots per edge * 4 edges = 8 dots.

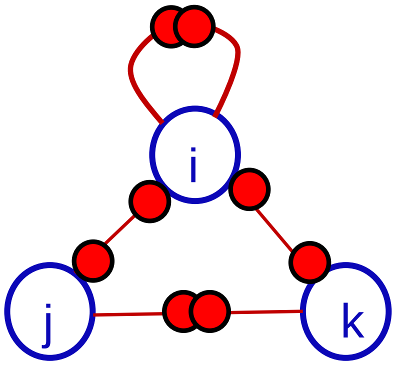
Now we move a dot to each vertex that is incident.



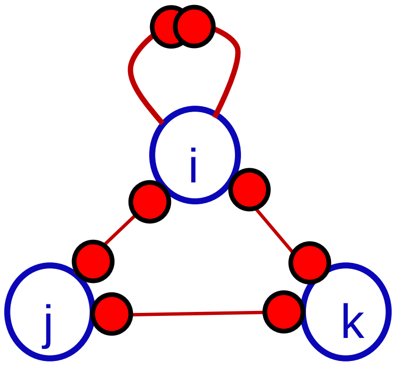
Now we move a dot to each vertex that is incident.



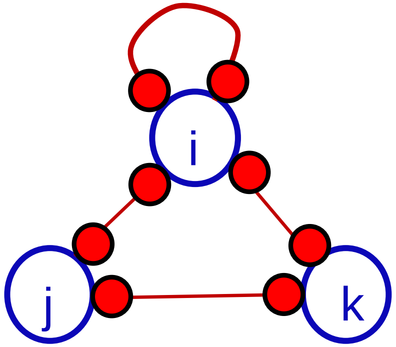
Now we move a dot to each vertex that is incident.



Now we move a dot to each vertex that is incident.



Now we move a dot to each vertex that is incident.



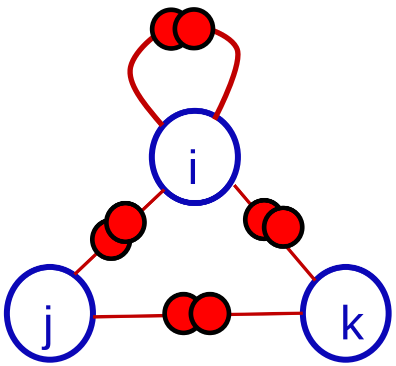
How many dots are there now? Still 8 dots.

How many dots are touching vertex i? 4 dots.

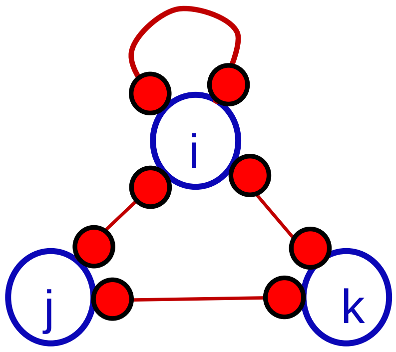
How many dots are touching vertex j? 2 dots.

How many dots are touching vertex k? 2 dots.

Depending on how we arrange the dots, we see them differently.



$2|E|$ = 8 dots.



$n_i = deg(\mathrm{i})$ = 4 dots touch i.

$n_j = deg(\mathrm{j})$ = 2 dots touch j.

$n_k = deg(\mathrm{k})$ = 2 dots touch k.

By accounting for the dots in two ways, we can see:

$$2|E| = 8 = 4 + 2 + 2 = deg(\mathrm{i}) + deg(\mathrm{j}) + deg(\mathrm{k}) = \sum_{u \in V} \deg(u) = \sum_{u\ in\ V} n_u$$

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