Lecture 13

CSE 331

Sep 25, 2024

Quiz 1 on Monday

note @126 stop following 0 views Actions

Quiz 1 on Monday, Sep 30

The first quiz will be from **11:00-11:10am in class** on **Monday, September 30**. We will have a 5 mins break after the quiz and the lecture will start at 11:15am.

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 T/F polls.) I will post sample mid-terms by Wednesday night so that you'll be able to see the formatting of such T/F questions.

Also quiz 1 will cover all topics we cover in class until Friday, Sep 27.

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 1 second ago by Atri Rudra

Busy 2 weeks ahead….

Mon, Sep 23	Breadth First Search F23 F22 F21 F19 F18 F17 x²	[KT, Sec 3.2]
Tue, Sep 24		(**HW 3 out, HW 2 in**)
Wed, Sep 25	Explore algorithm F23 F22 F21 F19 F118 F17 x²	[KT, Sec 3.3]
Fri, Sep 27	Runtime Analysis of BFS algorithm F23 F22 F21 F19 F18 F17 x²	[KT, Sec 3.3, 3.6] *Reading Assignment:* [KT, Sec 3.3, 3.4, 3.5, 3.6] *Reading Assignment:* Care package on topological ordering
Mon, Sep 30	More graph stuff F23 F22 F21 F19 F18 F17 x²	[KT, Sec 3.3, 3.6] (**Quiz 1**) (**Group Registration on Autolab due**)
Tue, Oct 1		(**HW 3 in**)
Wed, Oct 2	Interval Scheduling Problem F23 F22 F21 F19 F18 F17 x²	[KT, Sec 4.1] (**Project out**) *Reading Assignment:* [KT, Sec 4.1, 4.2]
Fri, Oct 4	Greedy Algorithm for Interval Scheduling F23 F22 F19 F18 F17 x²	[KT, Sec 4.4] *Reading Assignment:* Care package on minimizing maximum lateness
Mon, Oct 7	**Mid-term exam: I**	
Wed, Oct 9	**Mid-term exam: II**	

HW 3 out

Homework 3

Due by **11:30pm, Tuesday, October 1, 2024**.

Make sure you follow all the homework policies.

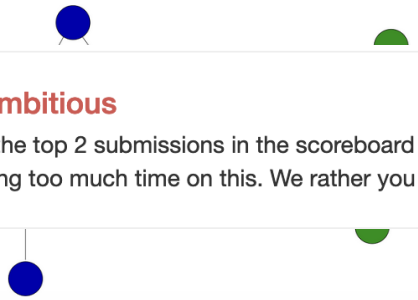
All submissions should be done via Autolab.

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).



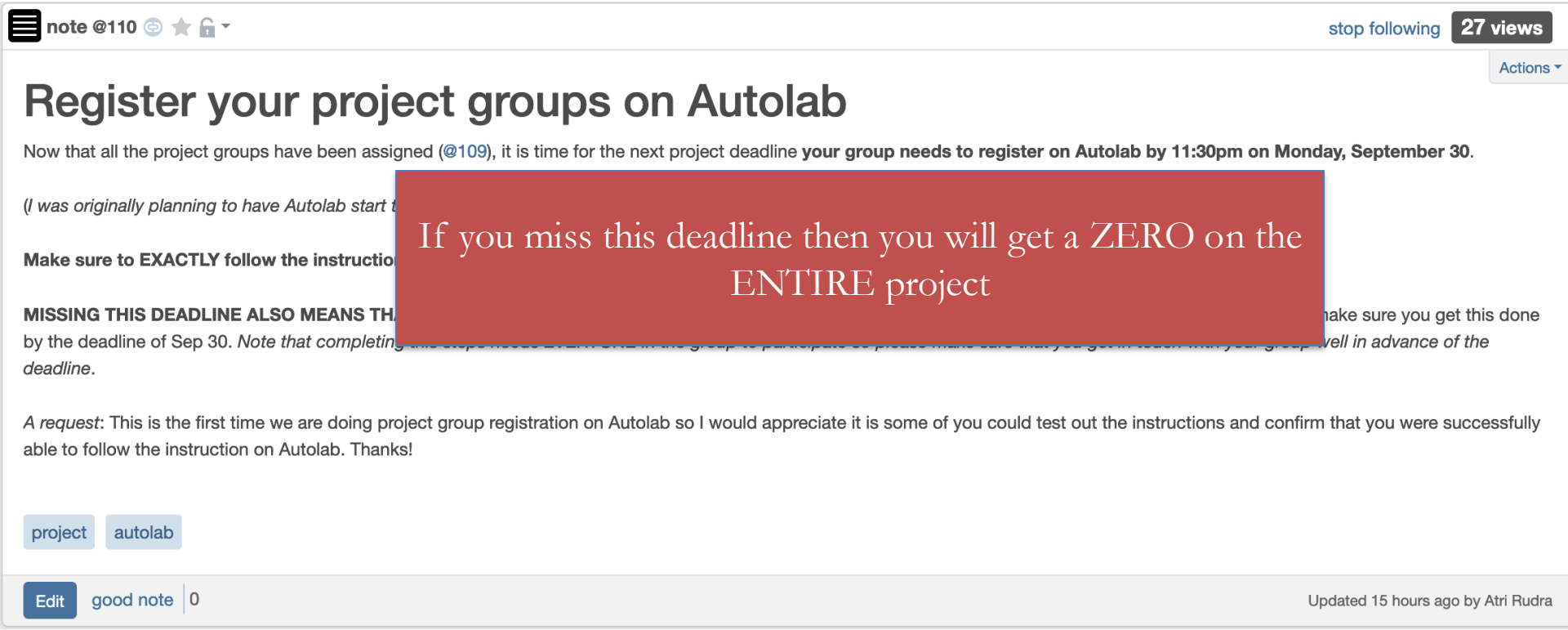
! For those of you who are feeling a little ambitious

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.

Note that a tree is a forest with 1 component.

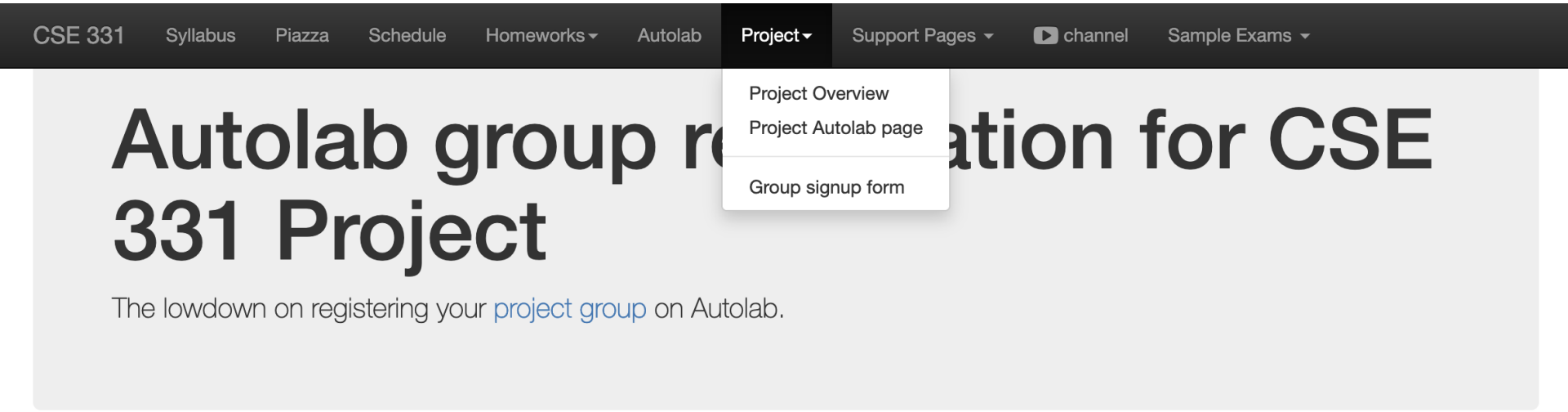
Autolab Project Group Registration

Also due Monday



If you miss this deadline then you will get a ZERO on the ENTIRE project

Read the instruction carefully



Fill in the Group Composition form FIRST

Make sure you fill in this Google form to submit your group composition. Please see the project overview page for more details on this.

You HAVE to submit the Google form

You **have** to submit this Google form by **11:30pm on Friday, September 20**. If you do not fill in the form on time, then you will not be able to register your group on Autolab and **will get a ZERO (0) on the ENTIRE project**.

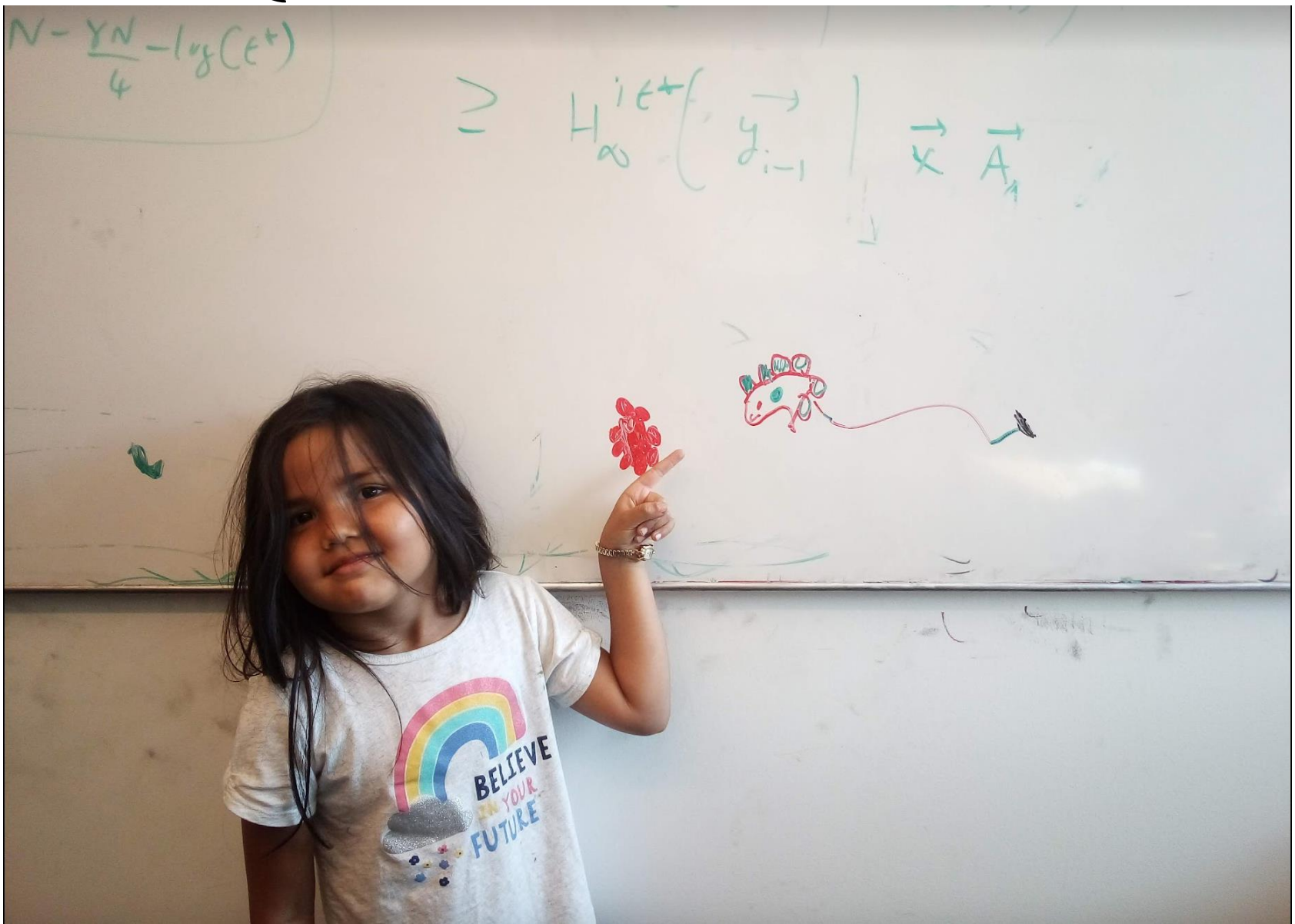
Register your group on Autolab

Groups on Autolab will NOT be automatically registered

Even after filling the Google form for your group composition you will have to register your group on Autolab by yourself (as a group). Read on for instructions on how to go about this.

However, once you register your group on Autolab you will **not** have to form your group for the coding and reflections submissions.

Questions/Comments?



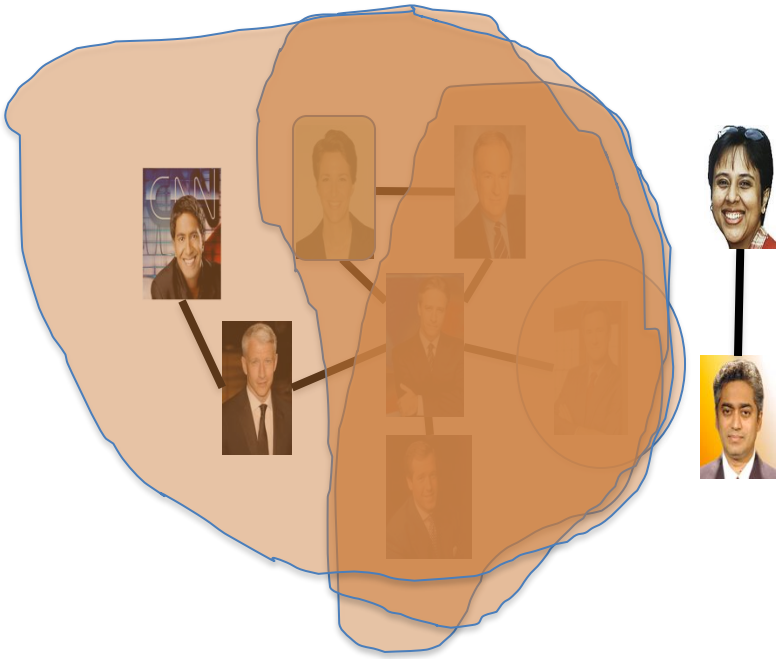
Connectivity Problem

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

Output: All t connected to s in G

Connected component of s also called CC(s)

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 $R^* = R$

BFS (Build layers of vertices)

$L_0 = \{s\}$

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

Explore is correct

Theorem: $CC(s) = R^*$

Lemma 1: $R^* \subseteq CC(s)$

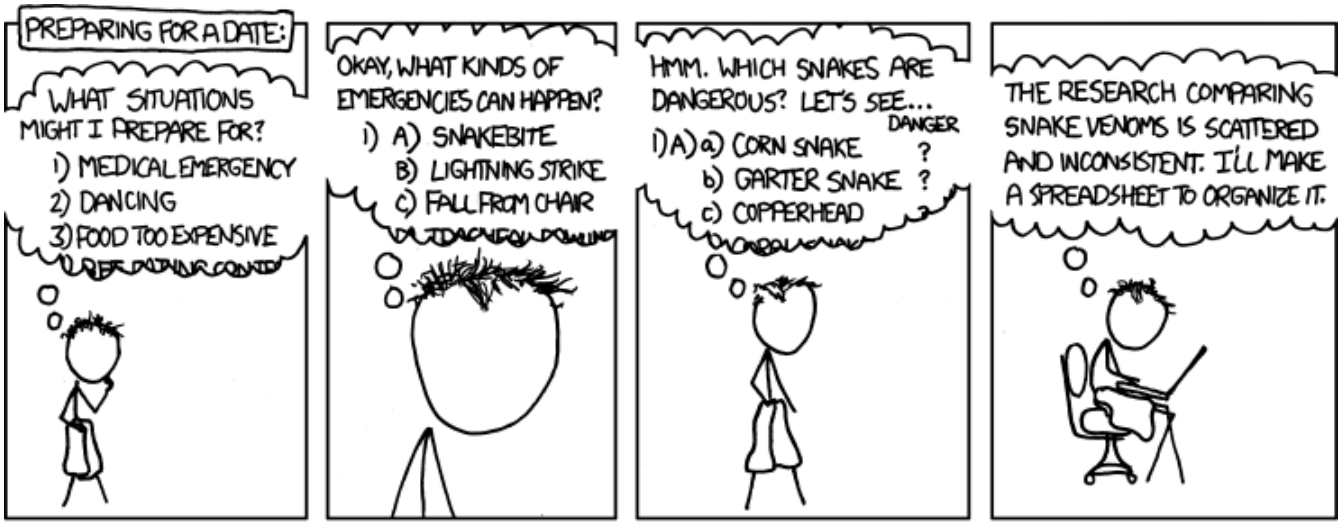
Lemma 2: $CC(s) \subseteq R^*$

Argue Lemma 2 on the board…

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/

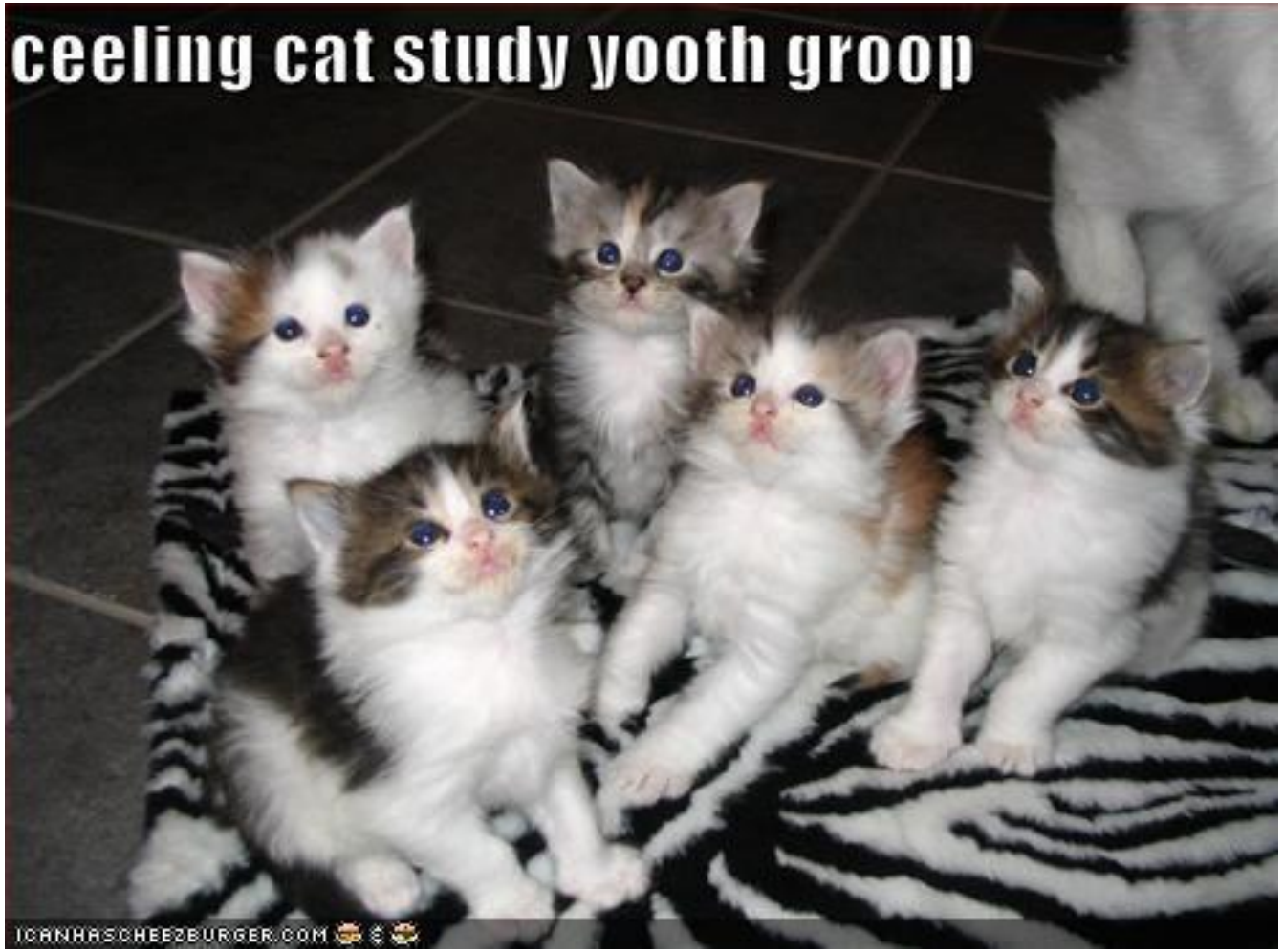
DFS(u)

Mark u as explored and add u to R

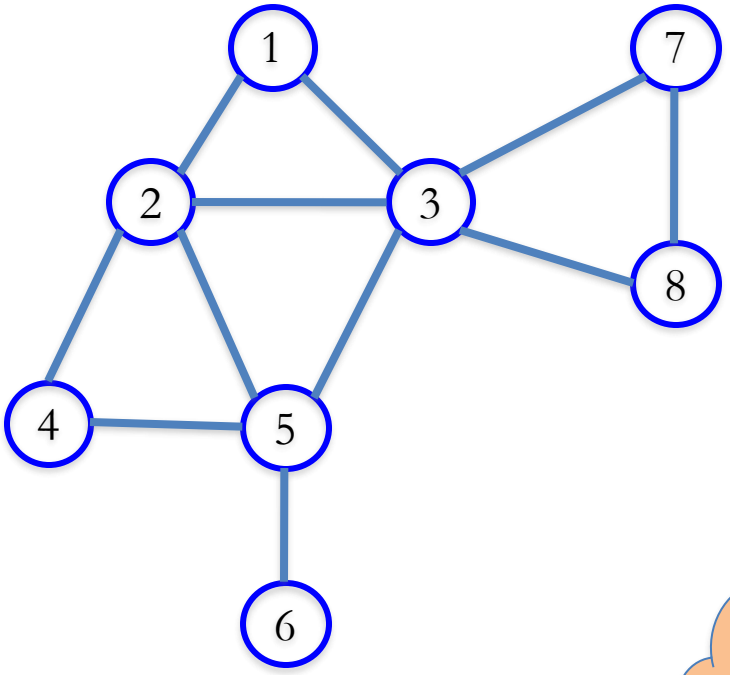
For each edge (u,v)

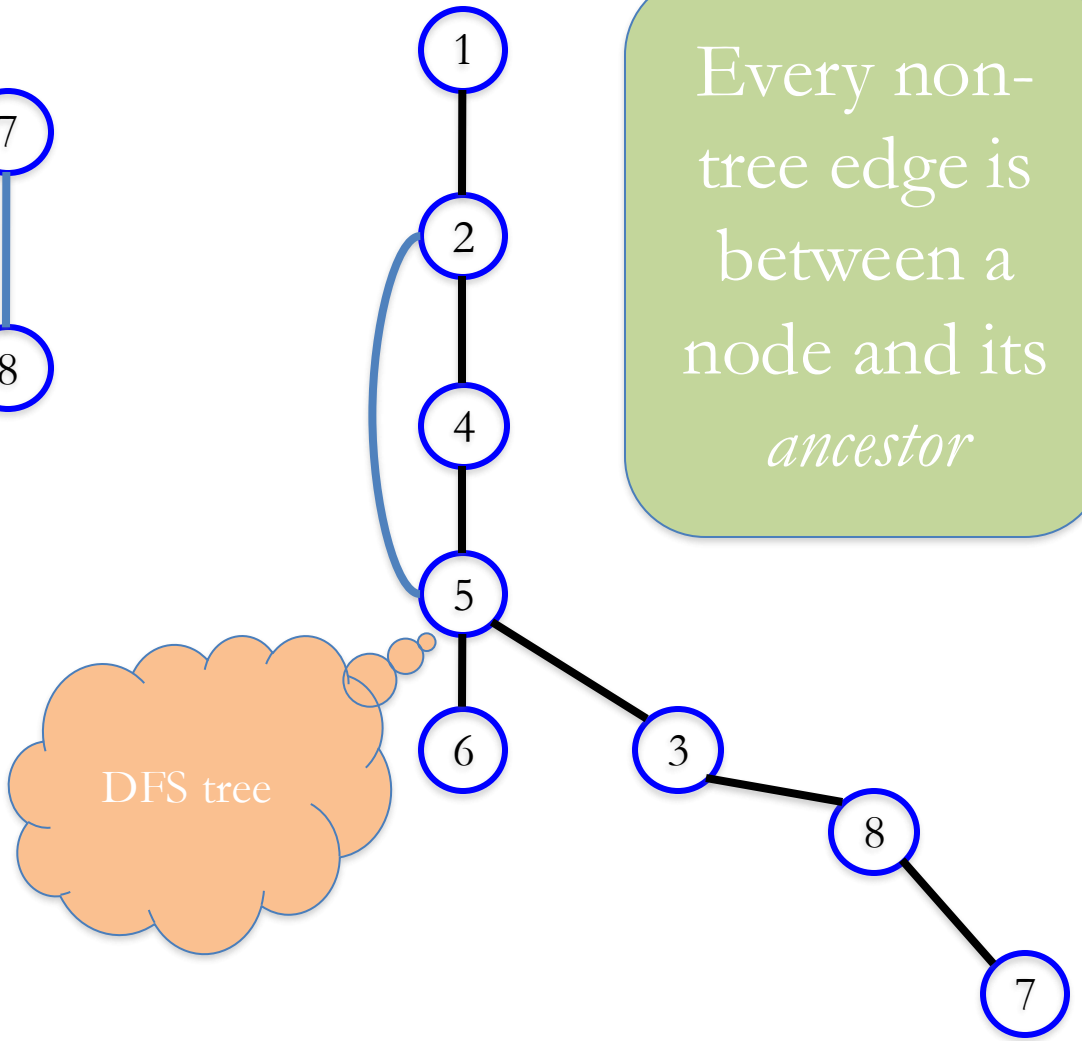
If v is not explored then DFS(v)

Why is DFS a special case of Explore?



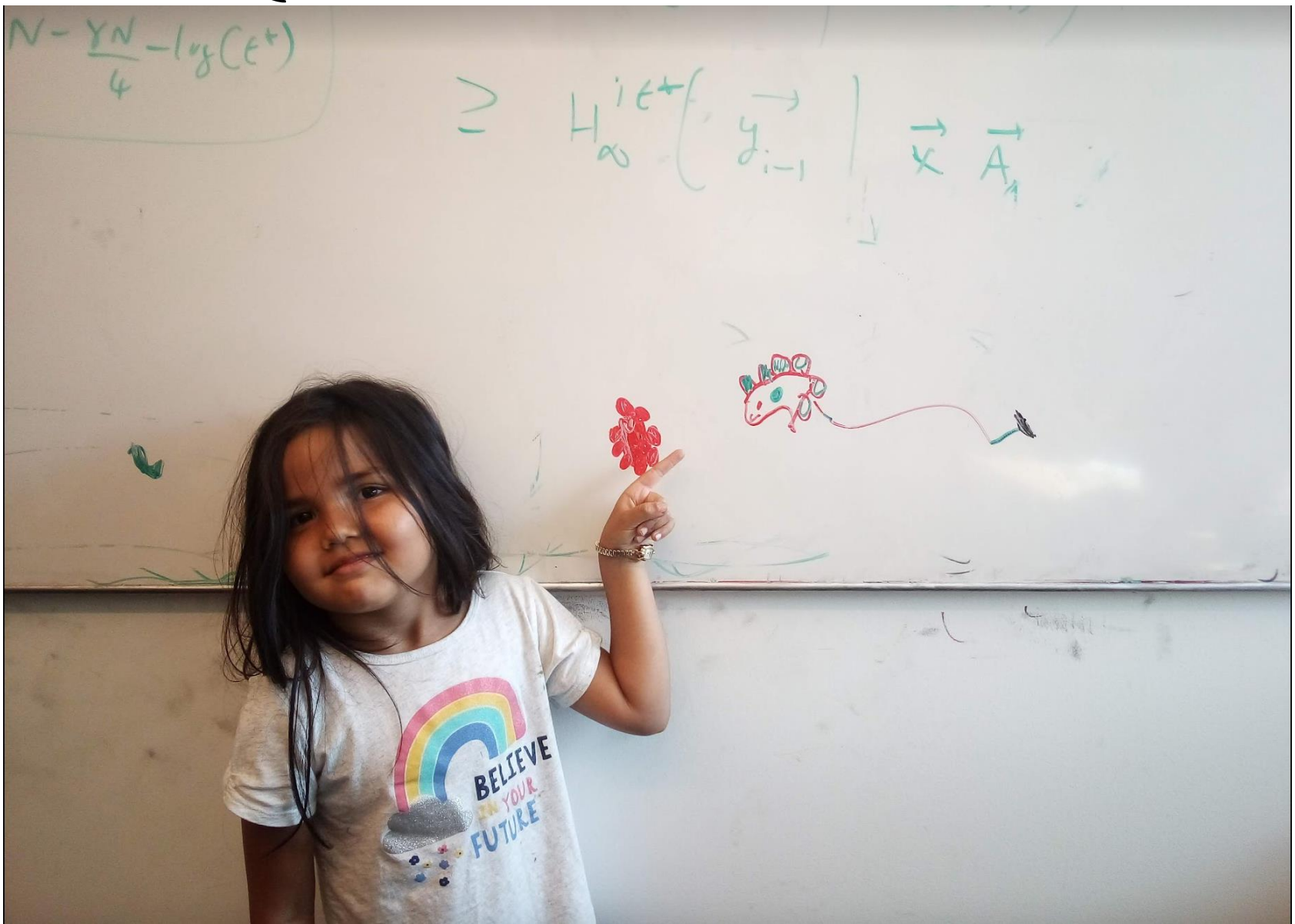
A DFS run





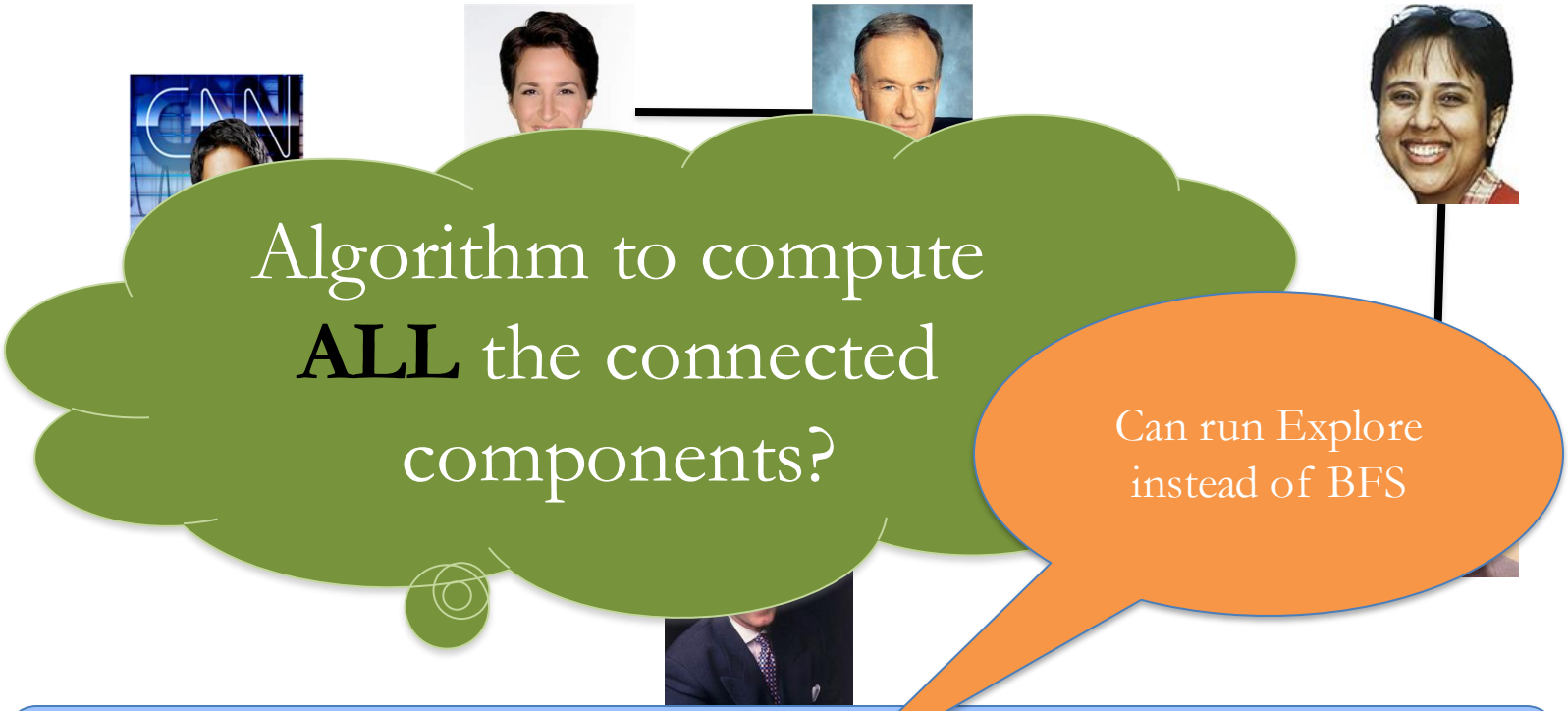
Every non-tree edge is between a node and its *ancestor*

Questions/Comments?

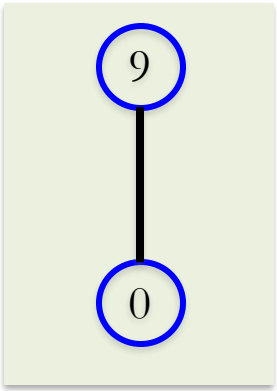


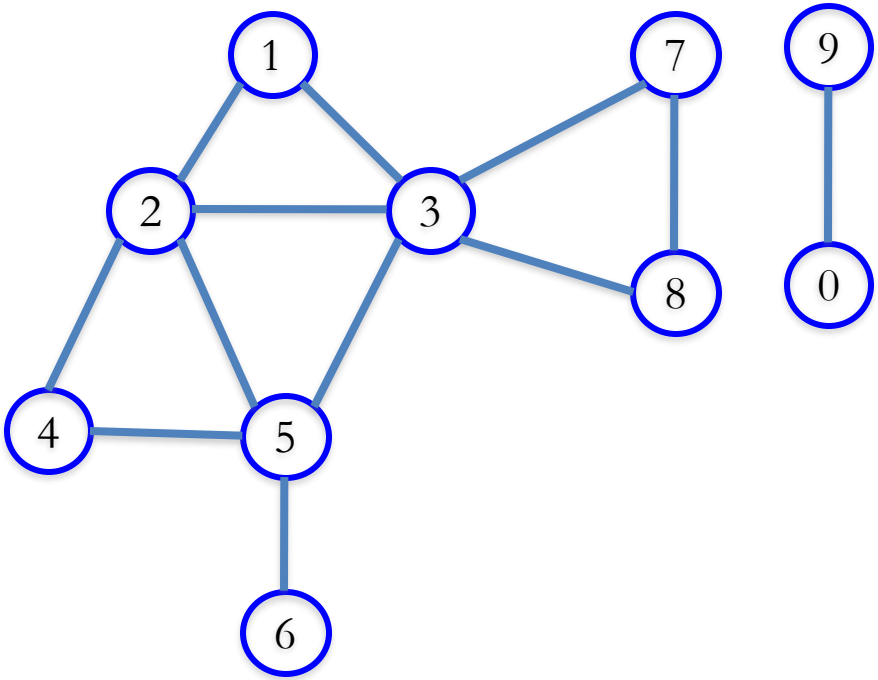
Connected components are disjoint

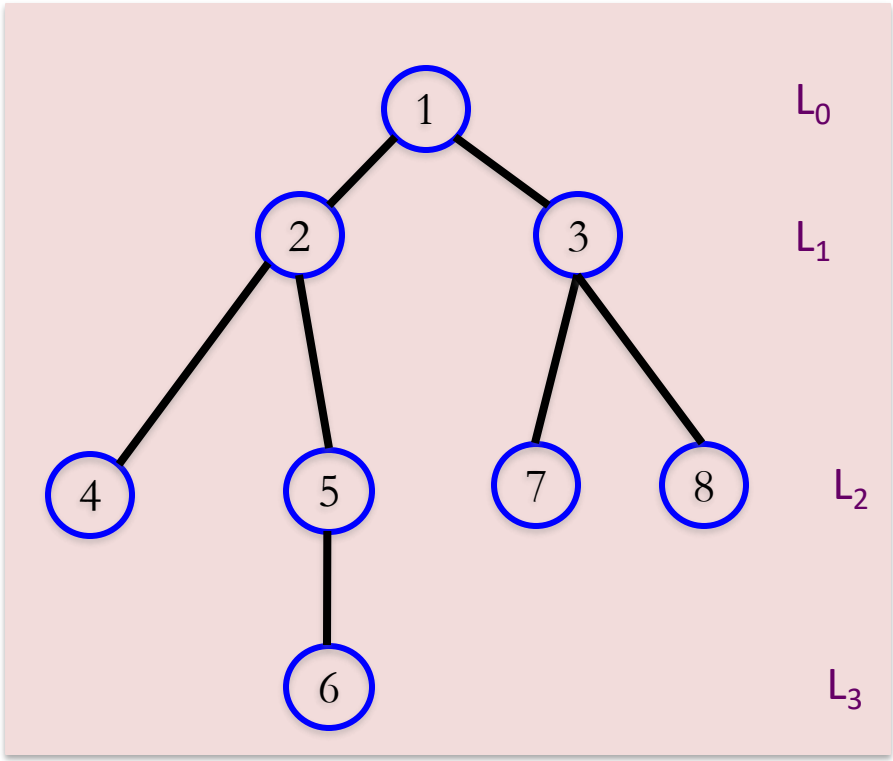
Either connected components of s and t are the same or are disjoint



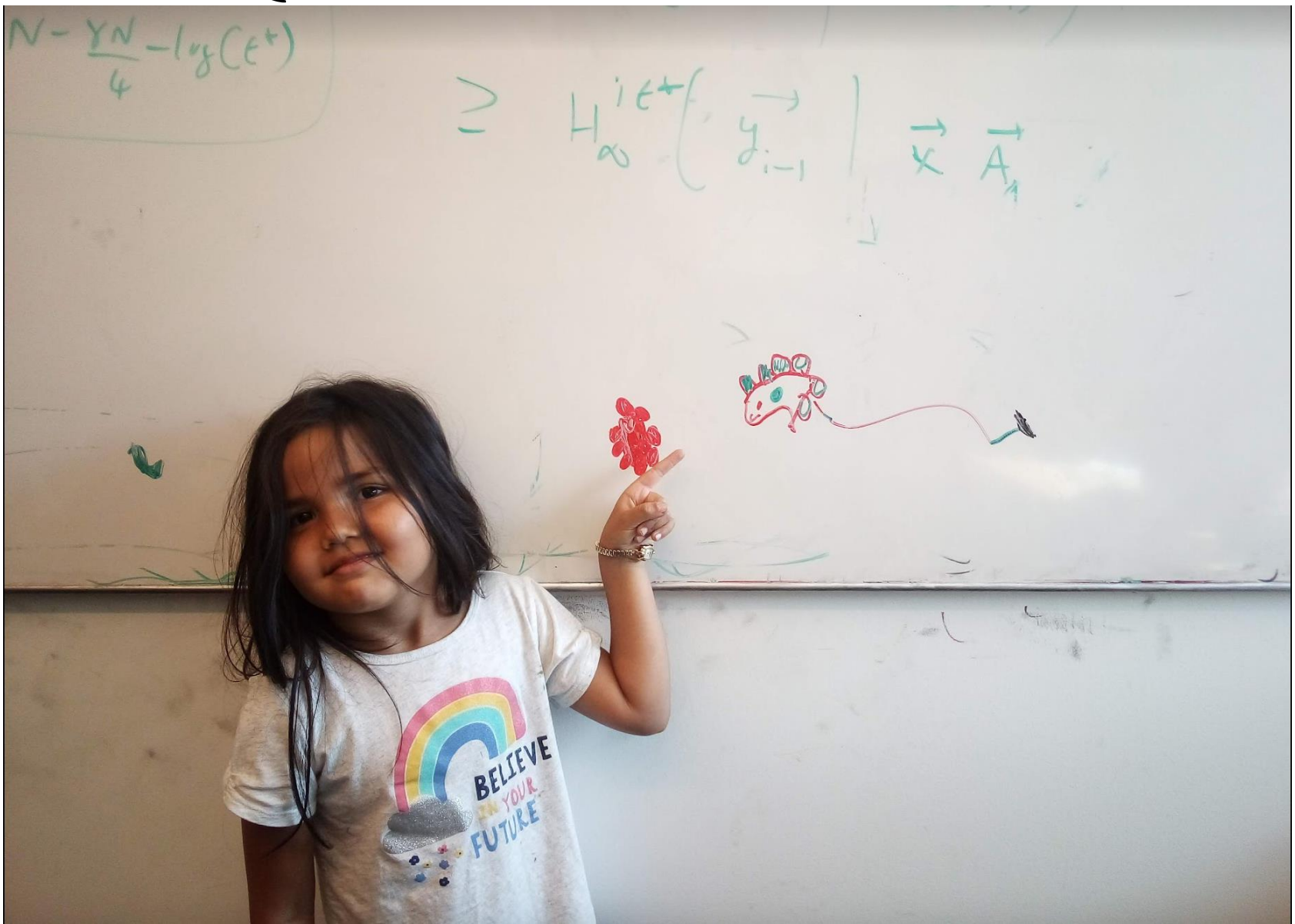
Computing all CCs







Questions/Comments?



Rest of today's agenda

Run-time analysis of BFS (DFS)

Stacks and Queues

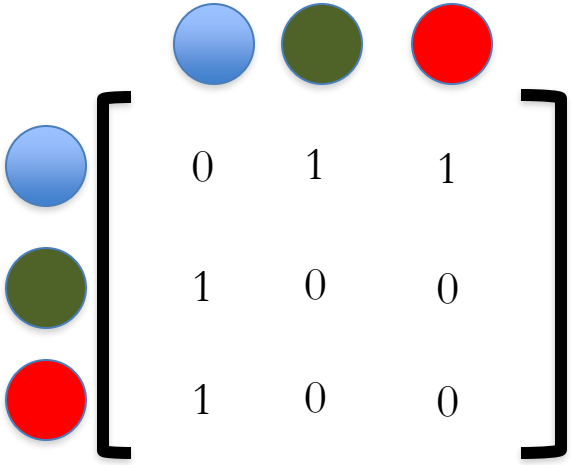
Last in First out

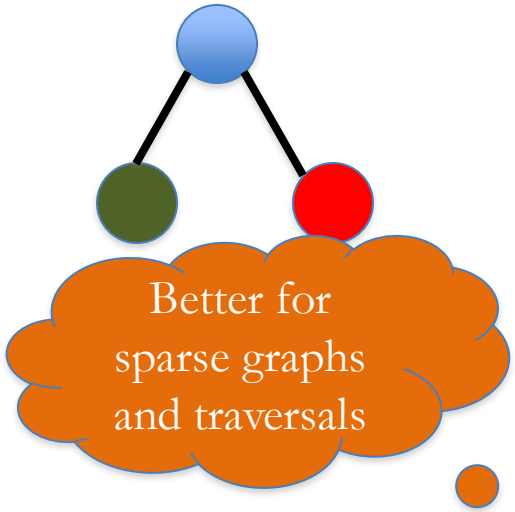
First in First out

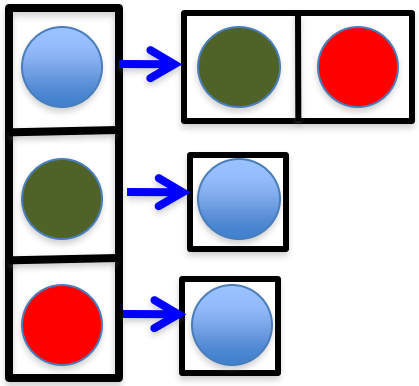
But first…

How do we represent graphs?

Graph representations

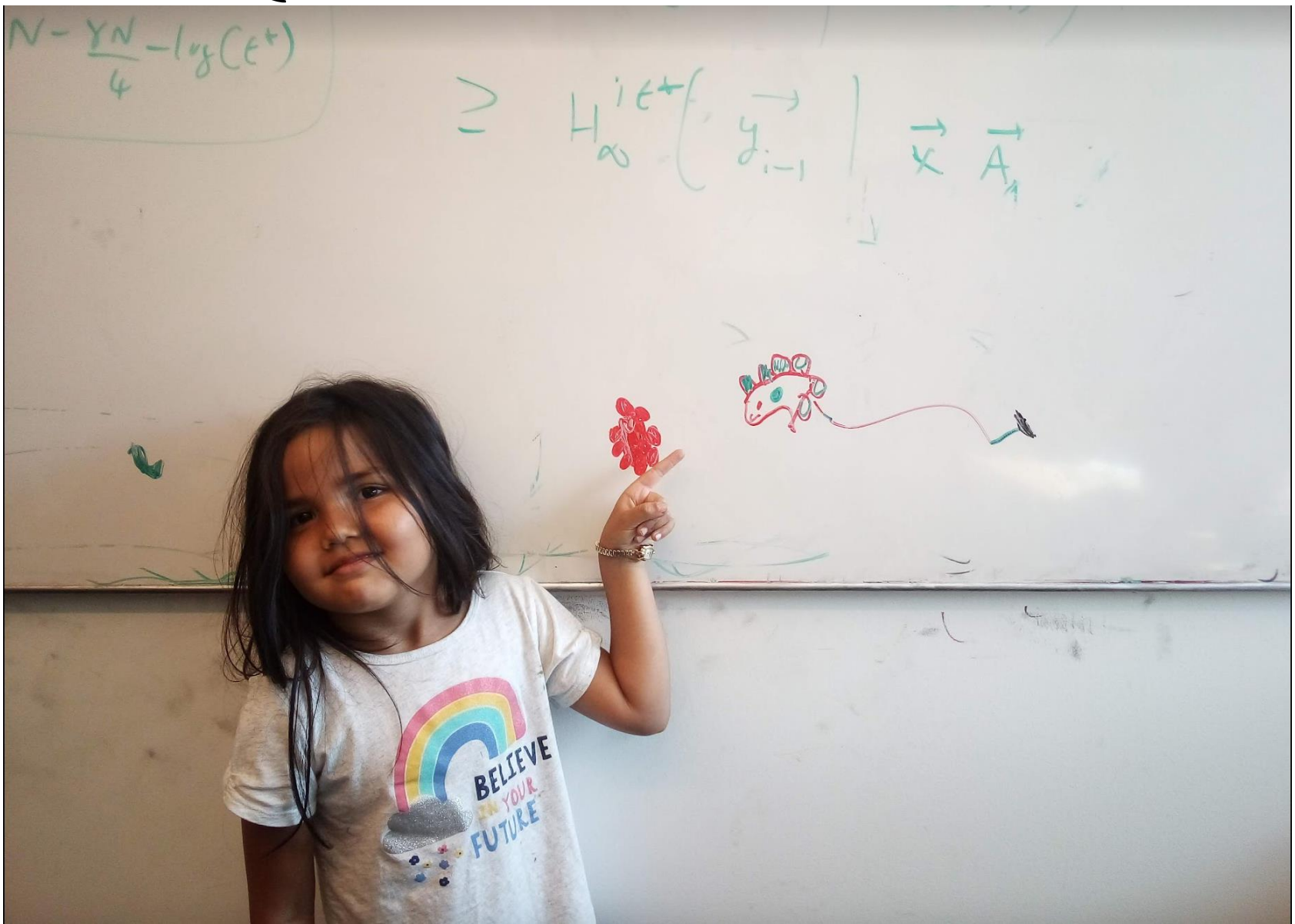






Adjacency matrix		Adjacency List
$O(1)$	(u,v) in E?	$O(n)$ [$O(n_v)$]
$O(n)$	All neighbors of u?	$O(n_u)$
$O(n^2)$	Space?	$O(m+n)$

Questions/Comments?

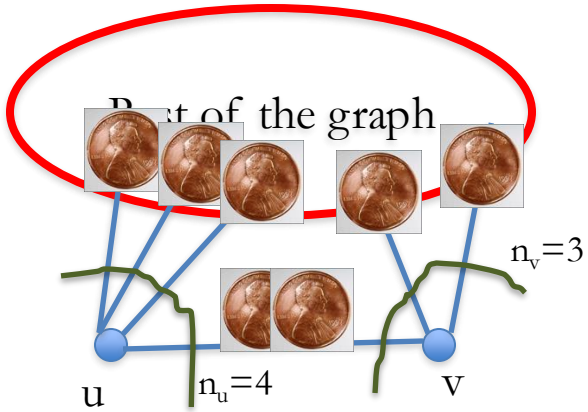


2 # edges = sum of # neighbors

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

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