Lecture 12

CSE 331

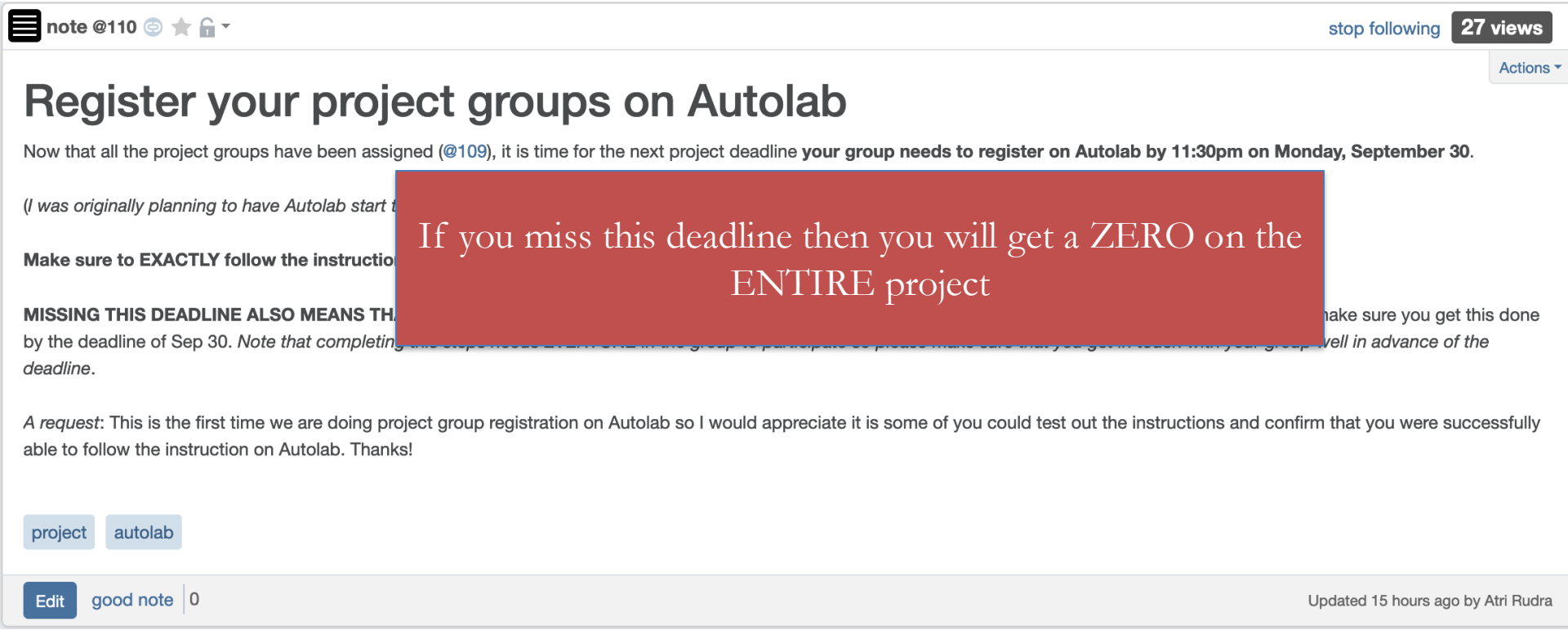
Sep 23, 2024

Quiz 1 next Monday

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**	

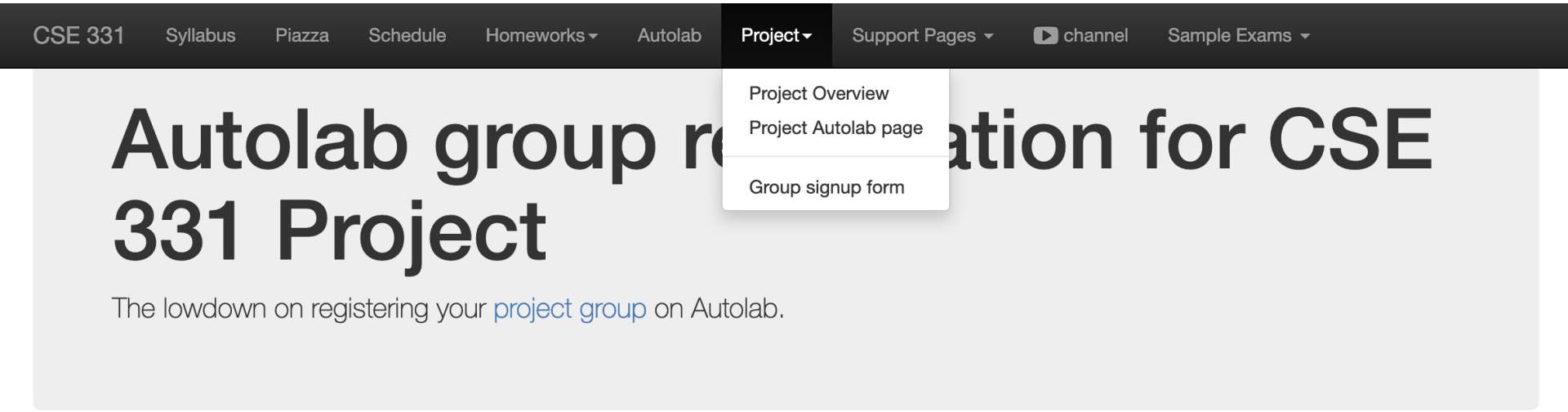
Autolab Project Group Registration

Also due next 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.

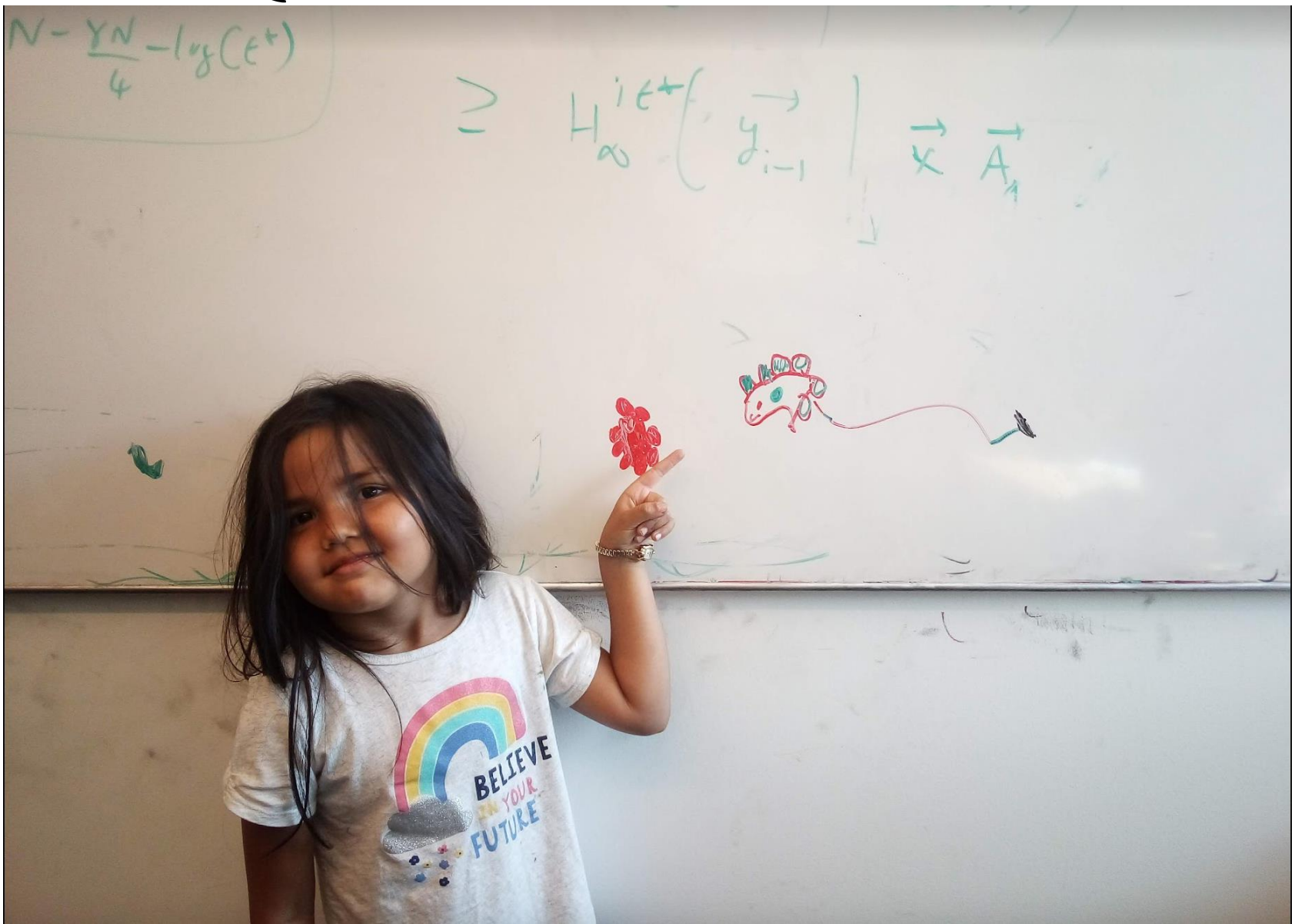
However, once you register your group on Autolab you will **not** have to form your group for the coding and reflections submissions.

Connectivity

u and w are connected iff there is a path between them

A graph is connected iff all pairs of vertices are connected

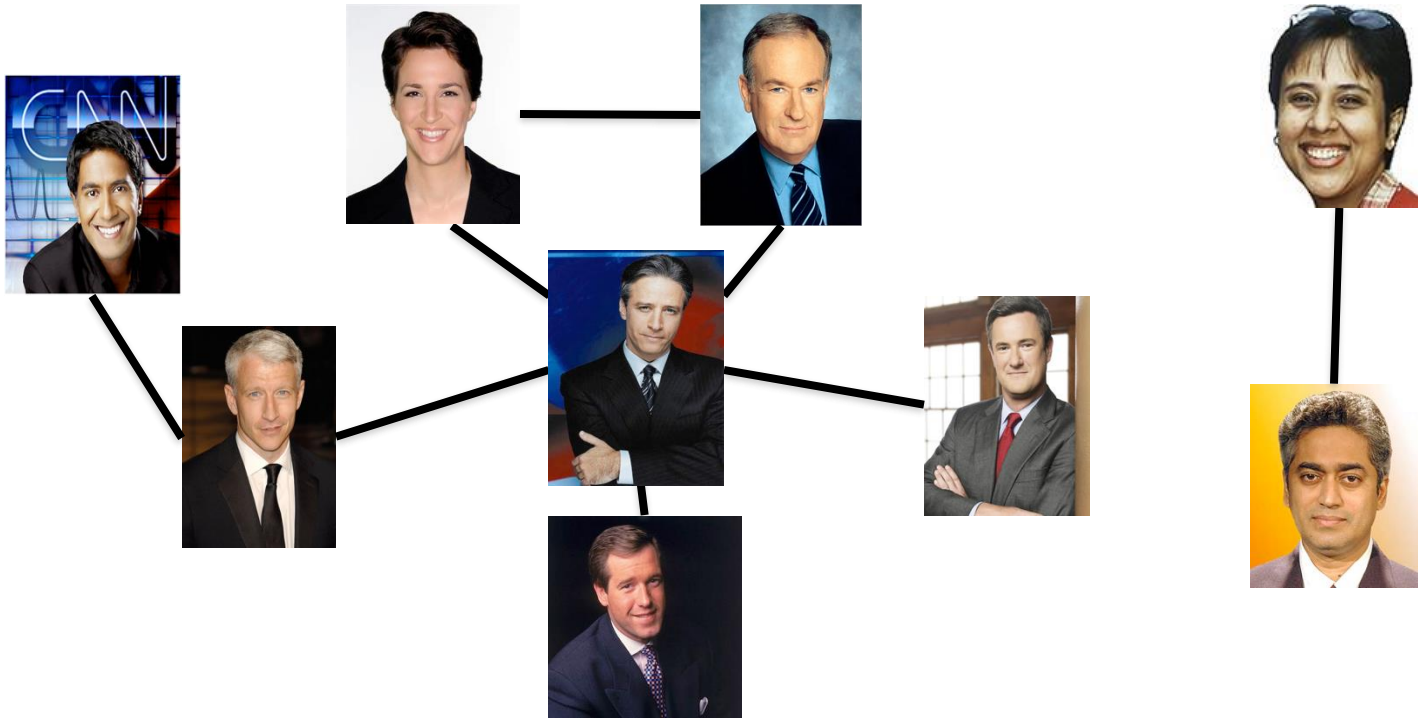
Questions/Comments?



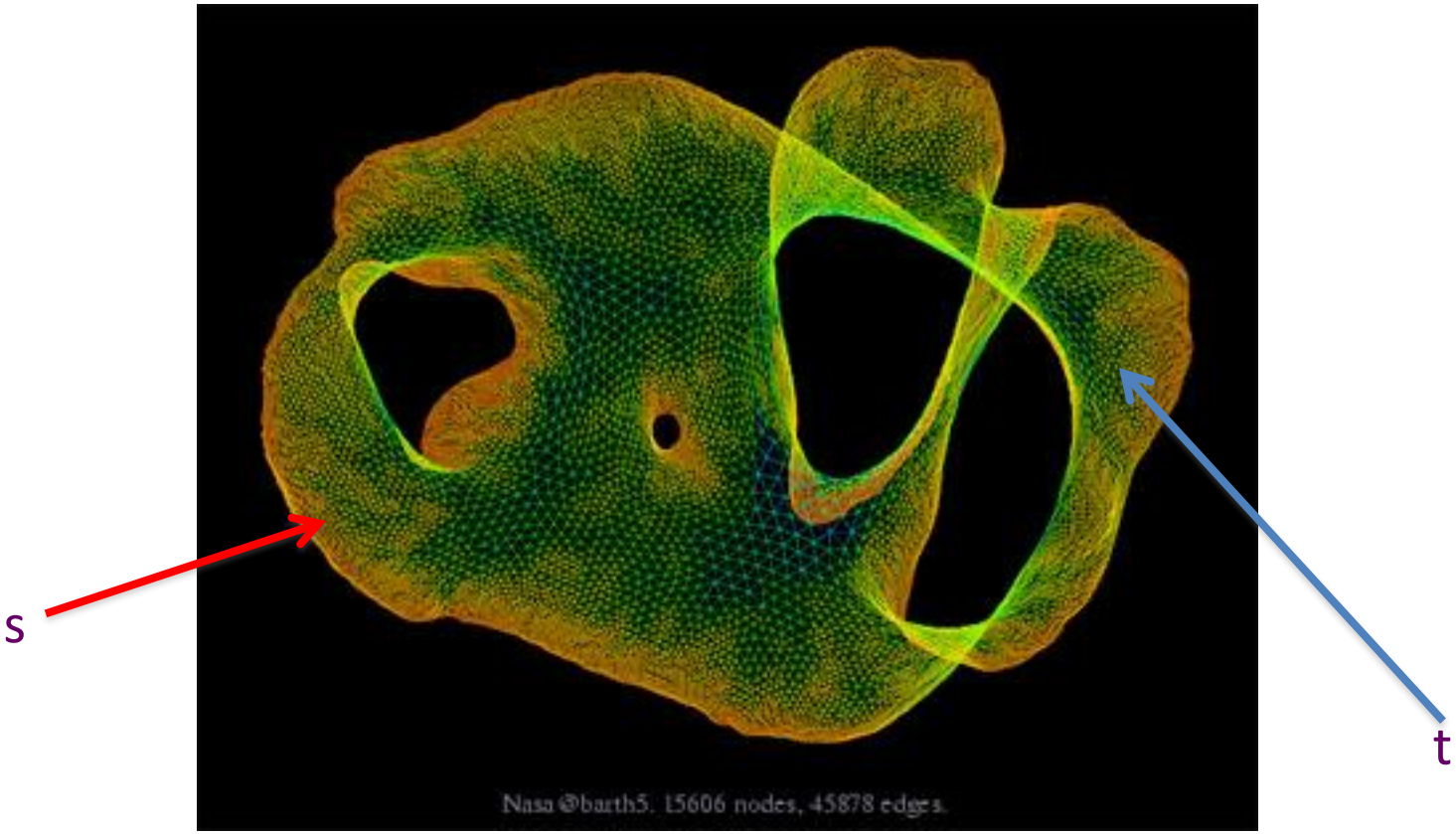
Rest of Today's agenda

Algorithms for checking connectivity

Checking by inspection



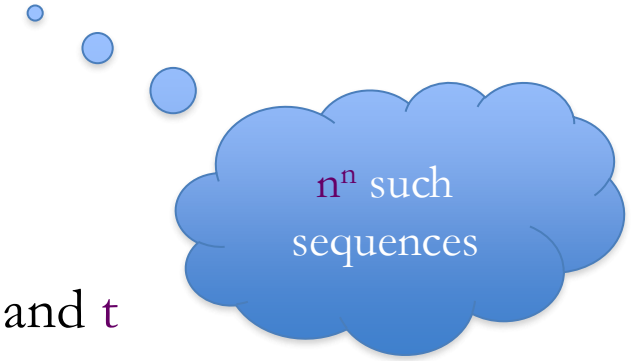
What about large graphs?



Are s and t connected?

Brute-force algorithm?

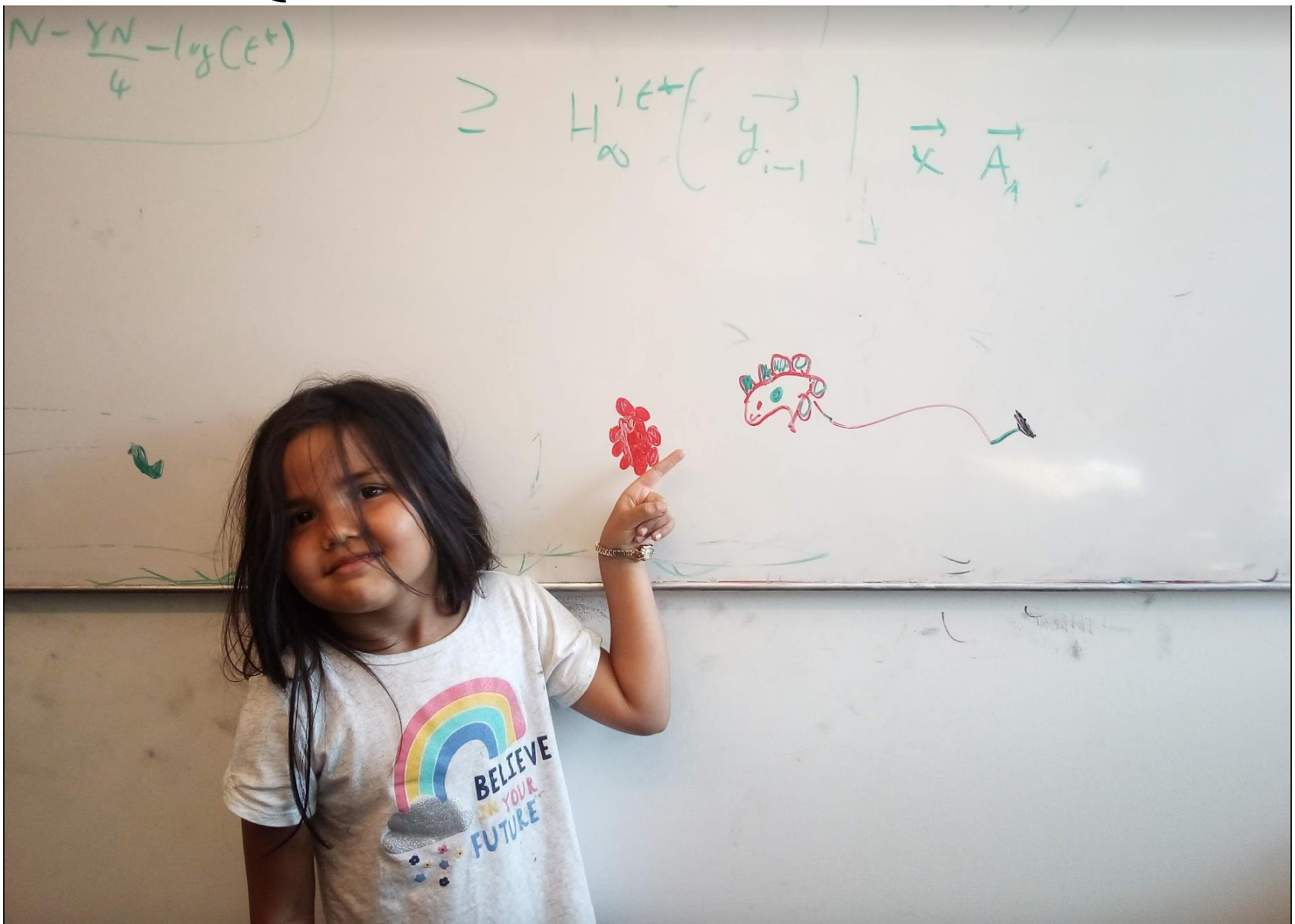
List all possible vertex sequences between s and t



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

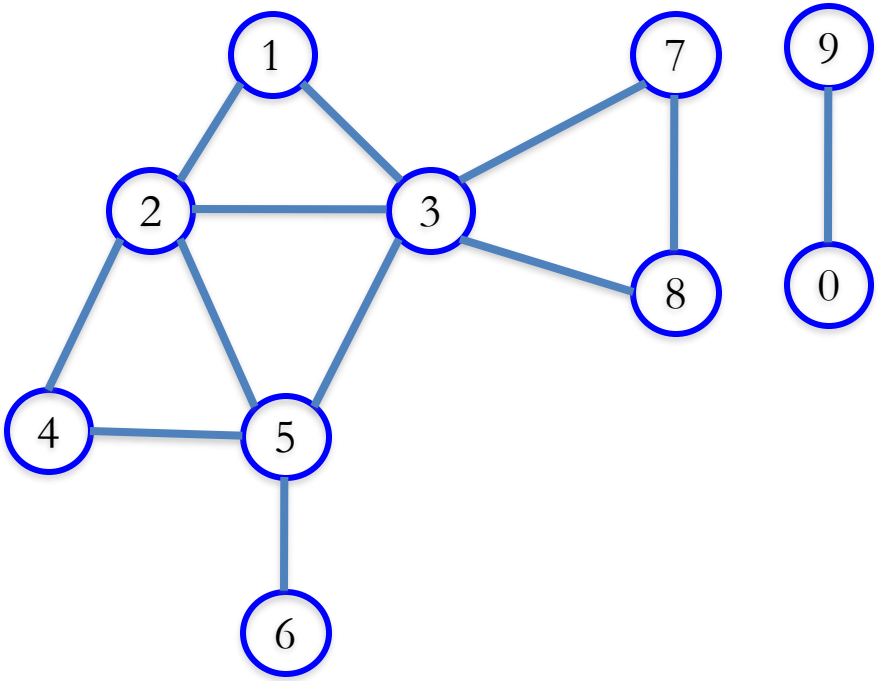
Stop when new layer is empty

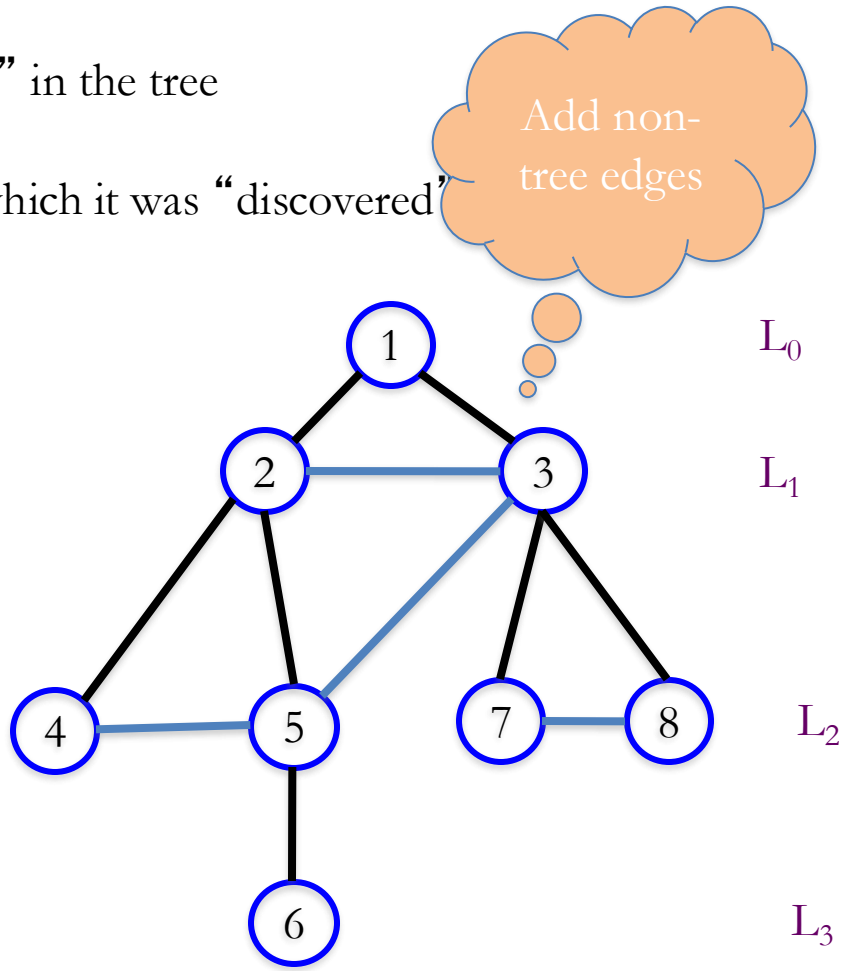
BFS Tree

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…

Two facts about BFS trees

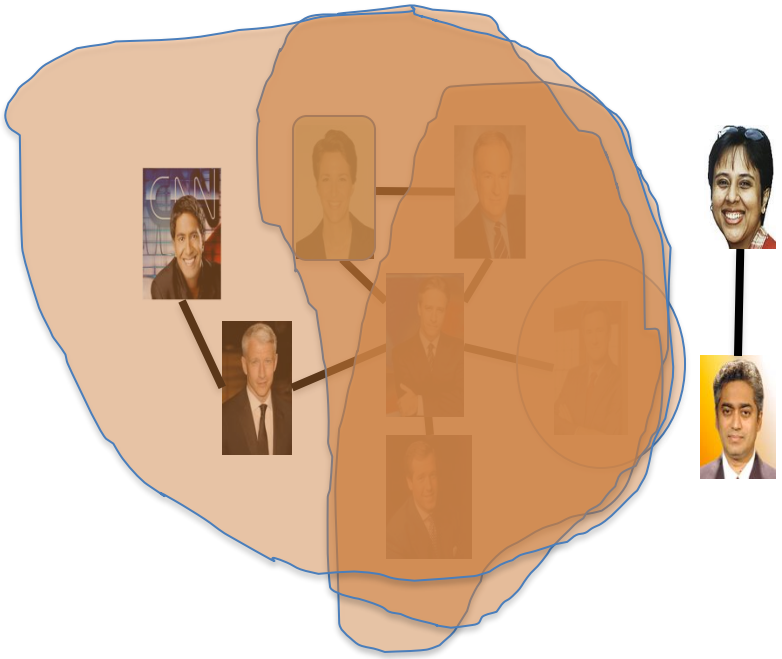
(1) All non-tree edges are in the same or consecutive layer

(2) If u is in L_i then $dist(s,u) = i$

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 $R^* = R$

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

Argue correctness on the board…