Lecture 12

CSE 331

Connectivity Problem

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

Output: All t connected to s in G

Algorithm motivation

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:

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

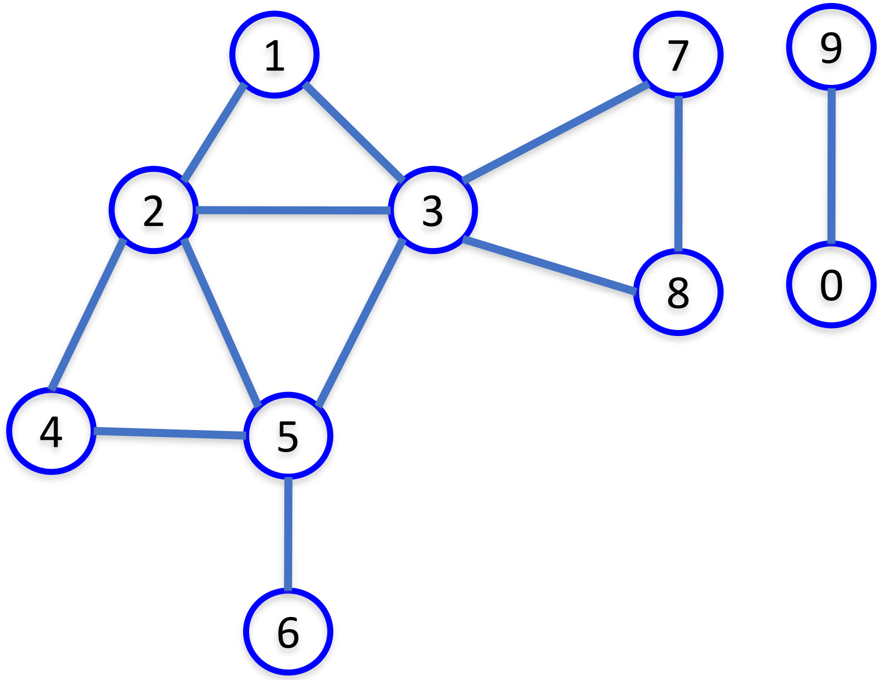
Stop when new layer is empty

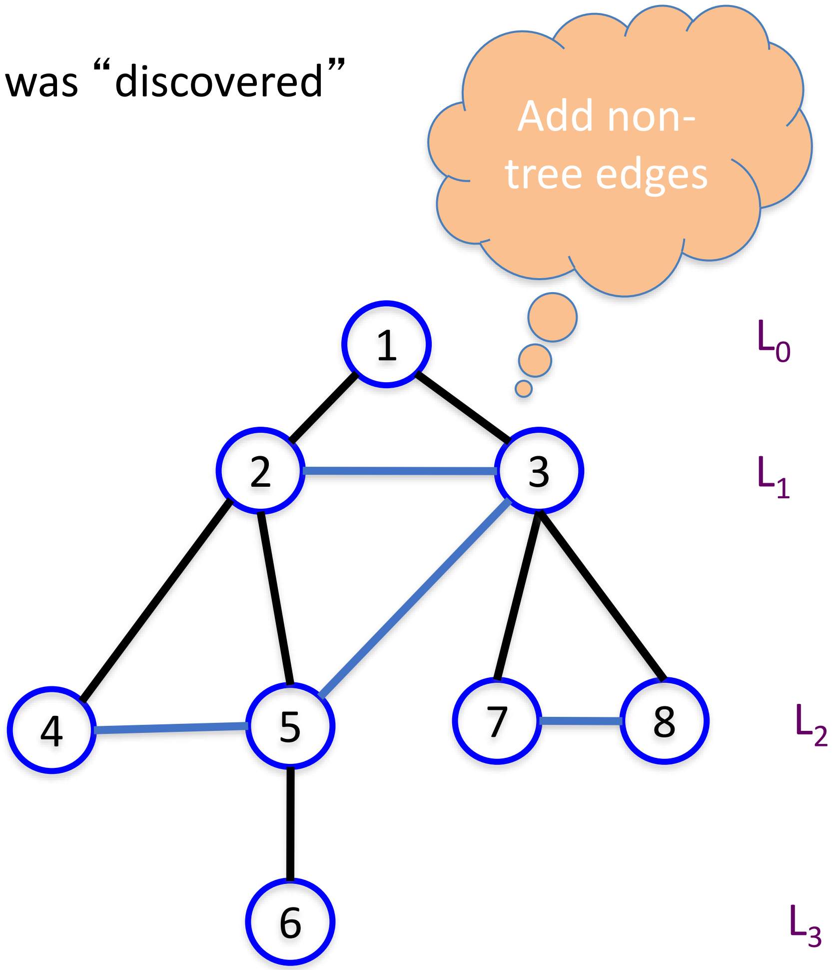
BFS Tree

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…

Connectivity Problem

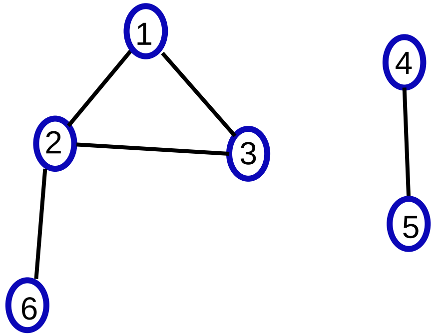
Input: Graph G = (V,E) and s in V

Output: All t connected to s in G

Connected component of s: CC(s)

Connected Components

Set of all vertices connected to vertex *s* is called its connected component cc(s).



cc(2) = {1, 2, 3, 6}

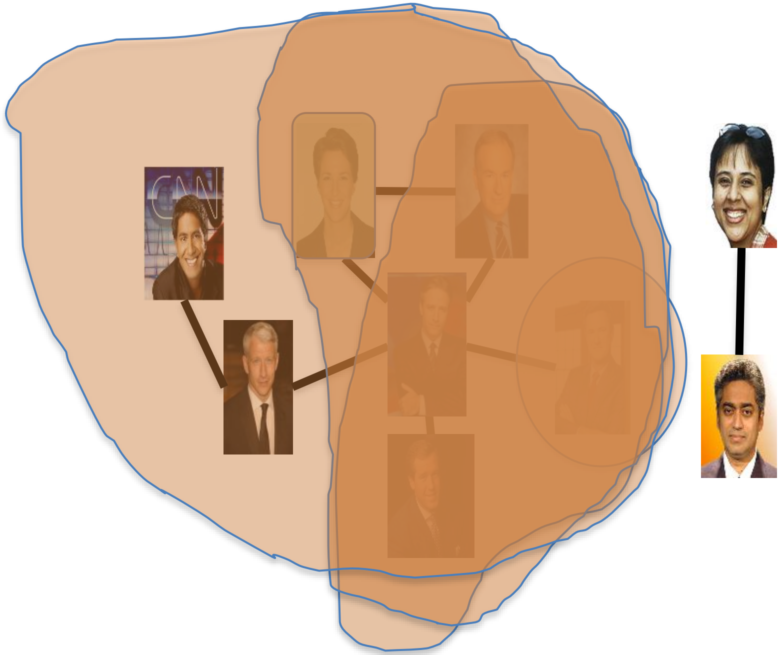
cc(5) = {4, 5}

Two facts about BFS trees

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

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

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…