Lecture 14

CSE 331

Sep 26, 2013

HW 3 due today

Place Q1, Q2 and Q3 in separate piles

I will not accept HWs after 1:15pm

Today's agenda

Computing Connected component (with DFS)

DFS(u)

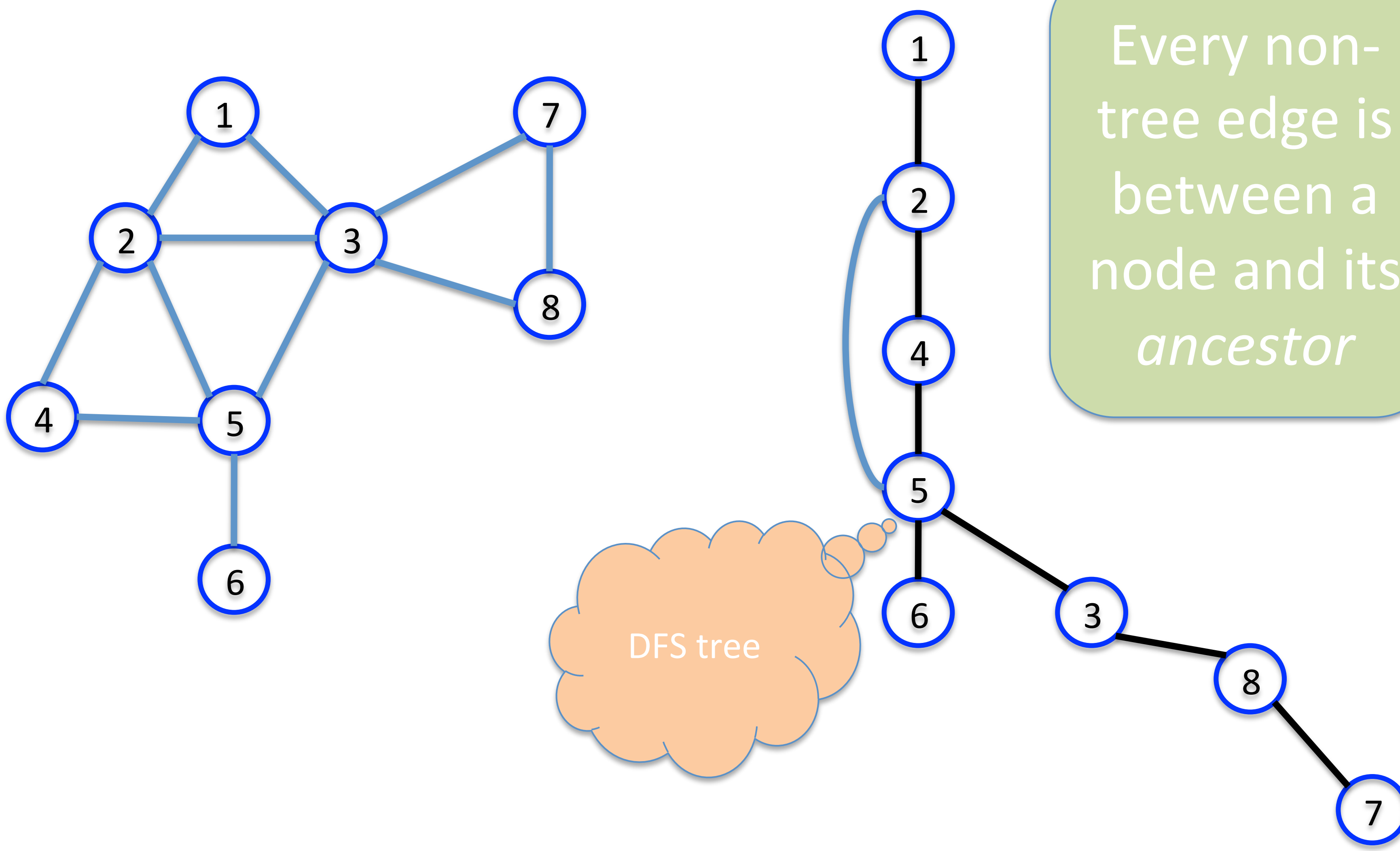
Mark u as explored and add u to R

For each edge (u,v)

If v is not explored then DFS(v)

Why is DFS a special case of Explore?

A DFS run

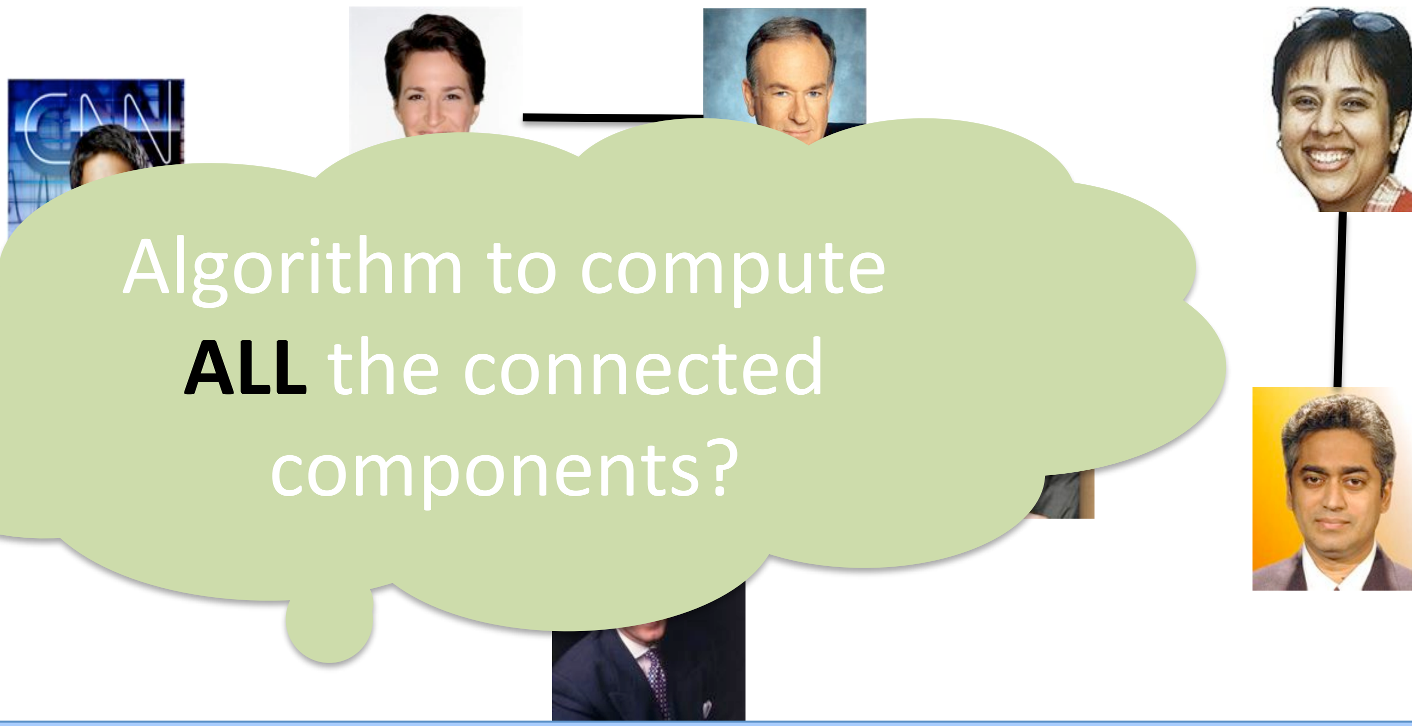


Every non-tree edge is between a node and its *ancestor*

Questions?

Connected components are disjoint

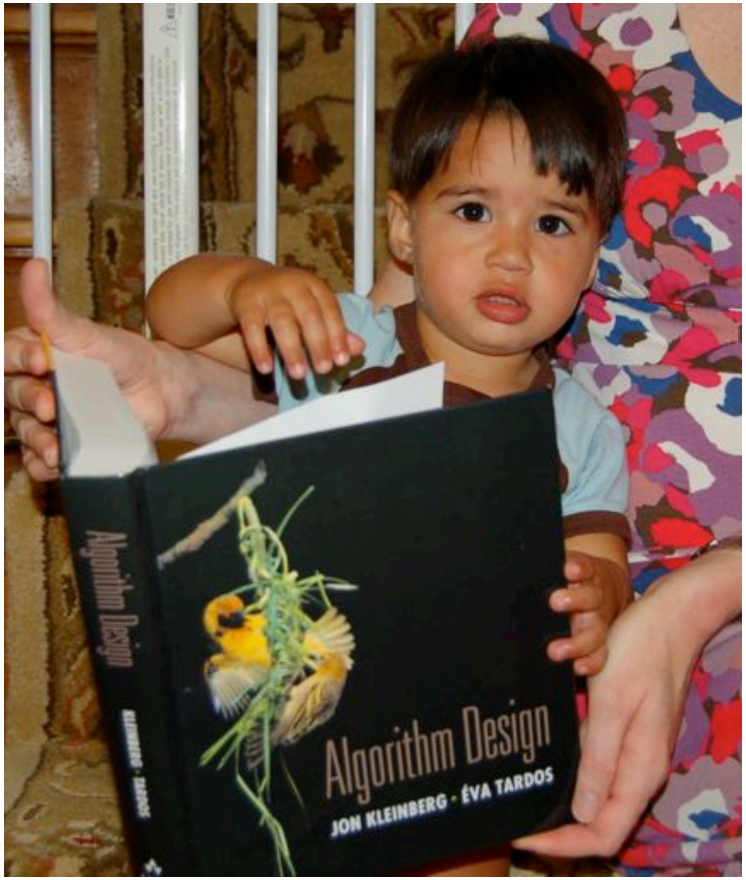
Either Connected components of s and t are the same or are disjoint



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

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

Reading Assignment



Sec 3.2 in [KT]

Rest of today's agenda

Run-time analysis of BFS (DFS)

Stacks and Queues

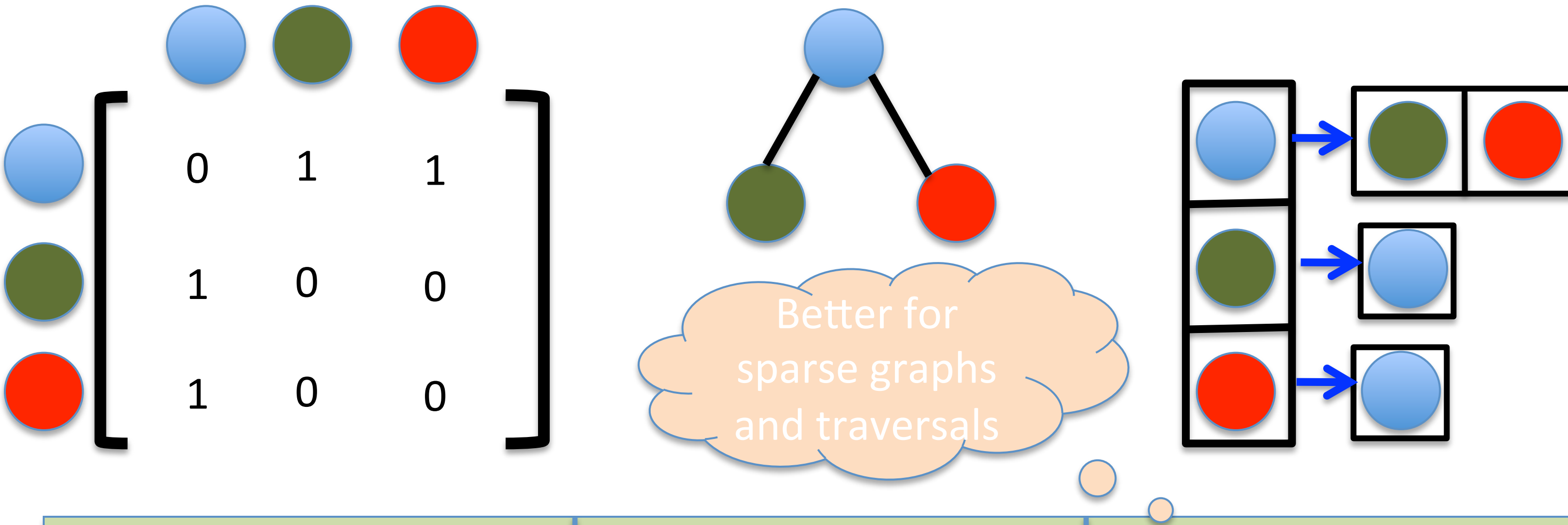
Last in First out

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)

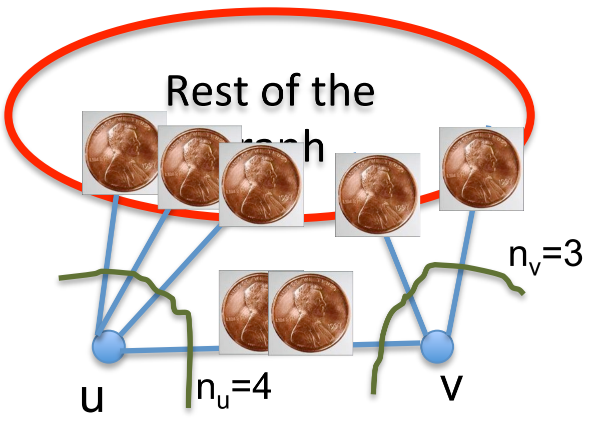
Questions?

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

An illustration