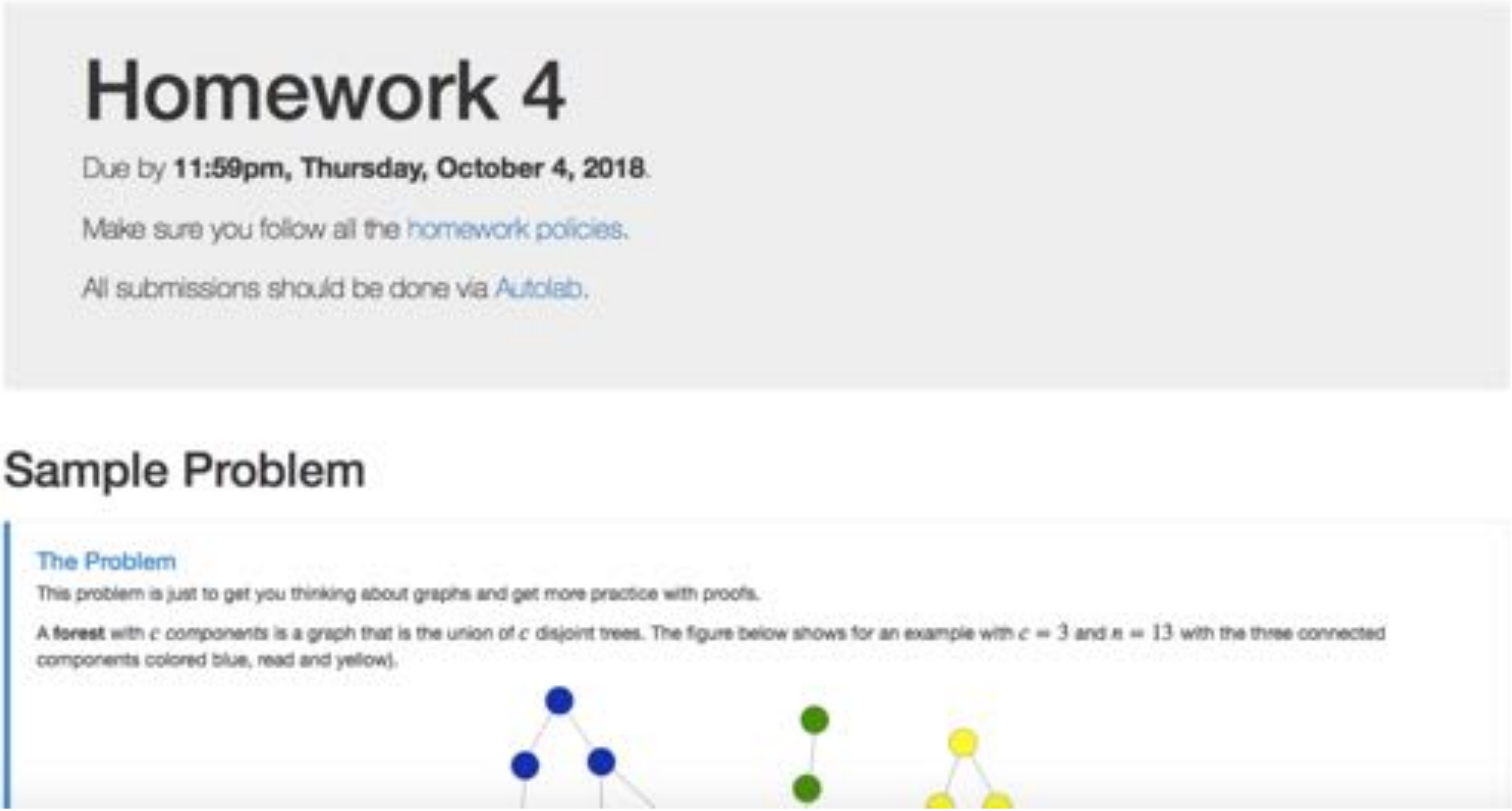
Lecture 14

CSE 331

Sep 28, 2018

HW 4 is now posted



Homework 4

Due by **11:59pm, Thursday, October 4, 2018**.

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).

Note: Bonus points for the fastest submissions on Q1. See **WARNING** though.

HW 3 solutions at end of lec

Today’s agenda

Run-time analysis of BFS (DFS)

Stacks and Queues

Last in First out

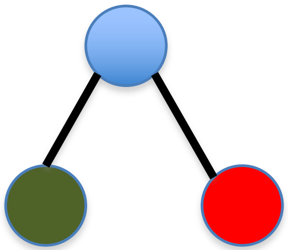
First in First out

But first…

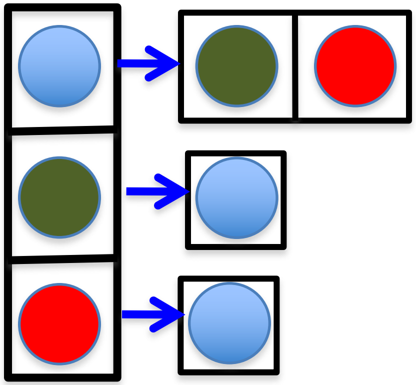
How do we represent graphs?

Graph representations

$$\begin{array}{c|ccc} & \text{🔵} & \text{🟢} & \text{🔴} \\ \hline \text{🔵} & 0 & 1 & 1 \\ \text{🟢} & 1 & 0 & 0 \\ \text{🔴} & 1 & 0 & 0 \end{array}$$



Better for sparse graphs and traversals

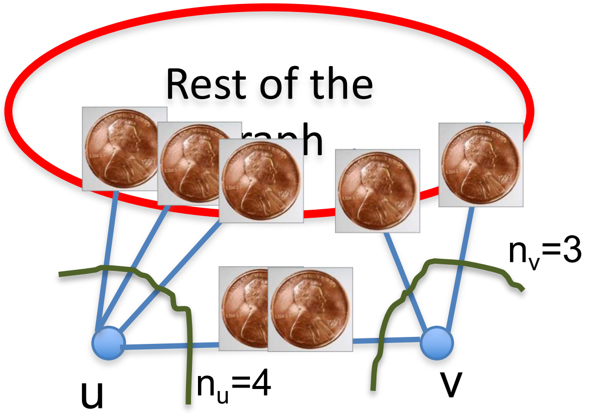


Adjacency matrix		Adjacency List
O(1)	(u,v) in E?	O(n) [O(n_v)]
O(n)	All neighbors of u?	O(n_u)
O(n^2)	Space?	O(m+n)

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

Rest of Today’s agenda

Space complexity of Adjacency list representation

Quick run time analysis for BFS

Quick run time analysis for DFS (and Queue version of BFS)

Helping you schedule your activities for the day