Lecture 11

CSE 331

Sep 22, 2016

Mini Project group due Monday!

note

0 views

Actions

Mini project needs groups of size EXACTLY 3

A gentle reminder that your group composition is due in just over a week (**11:59pm on Monday, Sep 26**).

The important thing to note is that you need to send me **groups of size EXACTLY three**. This means you are responsible for finding two other students in 331 to form your group. I will **not** make any group assignments.

Feel free to use the comments on this post to try and find others who are still looking to form a group.

mini_project

edit · good note | 0

Updated Just now by Atri Rudra

HW 3 is out!

CSE 331 Syllabus Piazza Schedule Homeworks ▾ Autolab Mini Project ▾ Support Pages ▾

Homework 3

Due by **12:30pm, Friday, September 30, 2016**.

Make sure you follow all the homework policies.

All submissions should be done via Autolab.

The support page for matrix vector multiplication should be very useful for this homework.

Sample Problem

The Problem

For this and the remaining problems, we will be working with $n \times n$ matrices (or two-dimensional arrays). So for example the following is a 3×3 matrix

$$\mathbf{M} = \begin{pmatrix} 1 & 2 & -3 \\ 2 & 9 & 0 \\ 6 & -1 & -2 \end{pmatrix}.$$

Support page is very imp.

CSE 331 Syllabus Piazza Schedule Homeworks Autolab Mini Project Support Pages

Matrix Vector Multiplication

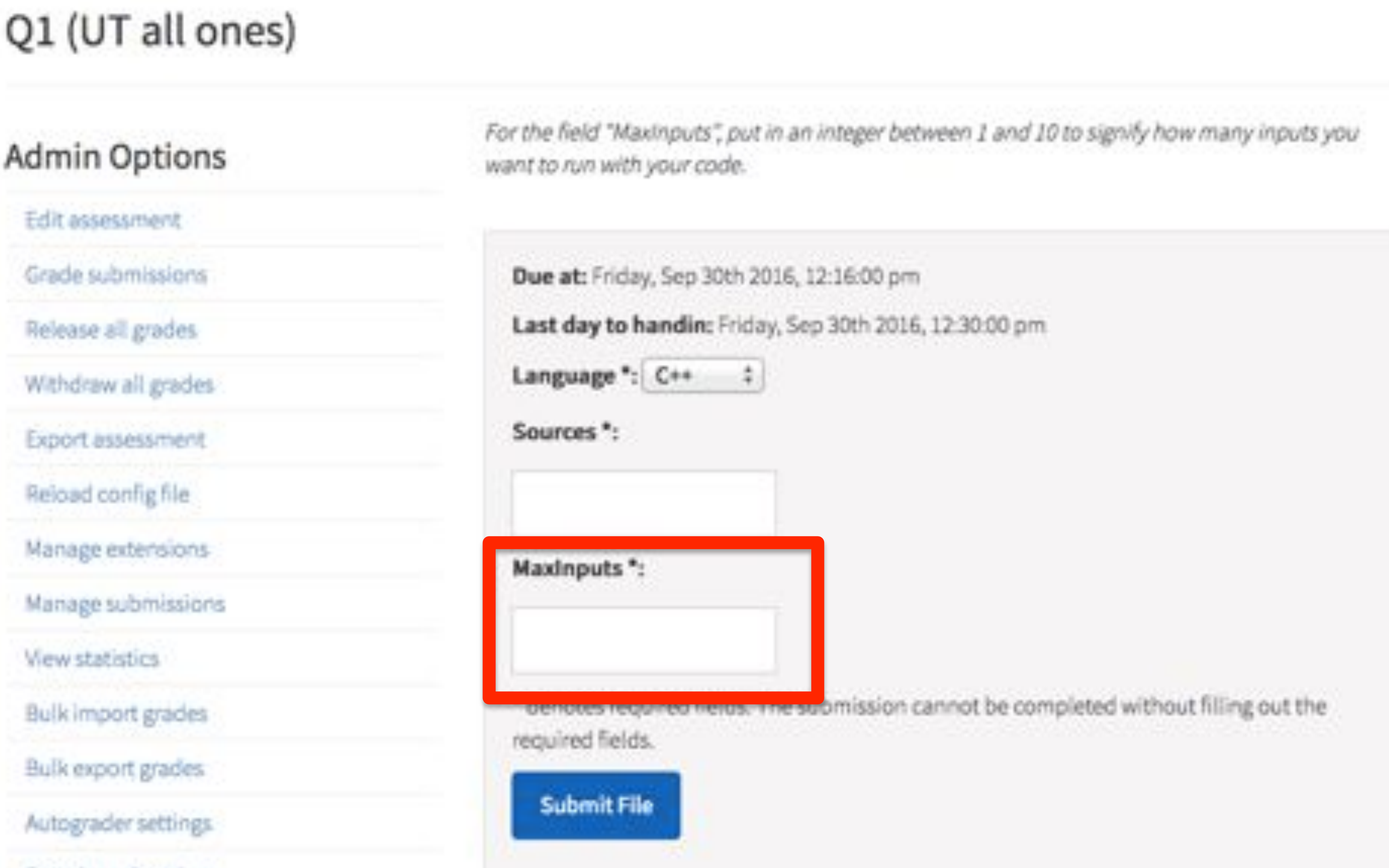
Martrix-vector multiplication is one of the most commonly used operations in real life. We unfortunately won't be able to talk about this in CSE 331 lectures, so this page is meant as a substitute. We will also use this as an excuse to point out how a very simple property of numbers can be useful in speeding up algorithms.

Background

In this note we will be working with matrices and vectors. Simply put, matrices are two dimensional arrays and vectors are one dimensional arrays (or the "usual" notion of arrays). We will be using notation that is consistent with array notation. So e.g. a matrix A with m rows and n columns (also denoted as an $m \times n$ matrix) will in code be defined as `int [][] A = new int[m][n]` (assuming the matrix stores integers). Also a vector x of size n in code will be declared as `int [] x = new int[n]` (again assuming the vector contains integers). To be consistent with the array notations, we will denote the entry in A corresponding to the ith row and jth column as $A[i][j]$ (or `A[i][j]`). Similarly, the ith entry in the vector x will be denoted as $x[i]$ (or `x[i]`). We will follow the array convention assume that the indices i and j start at 0.

If you want a refresher on matrices, you might want to start with this Khan academy video (though if you are comfortable with the array analogy above you should not really need much more for this note):

New autograding feature

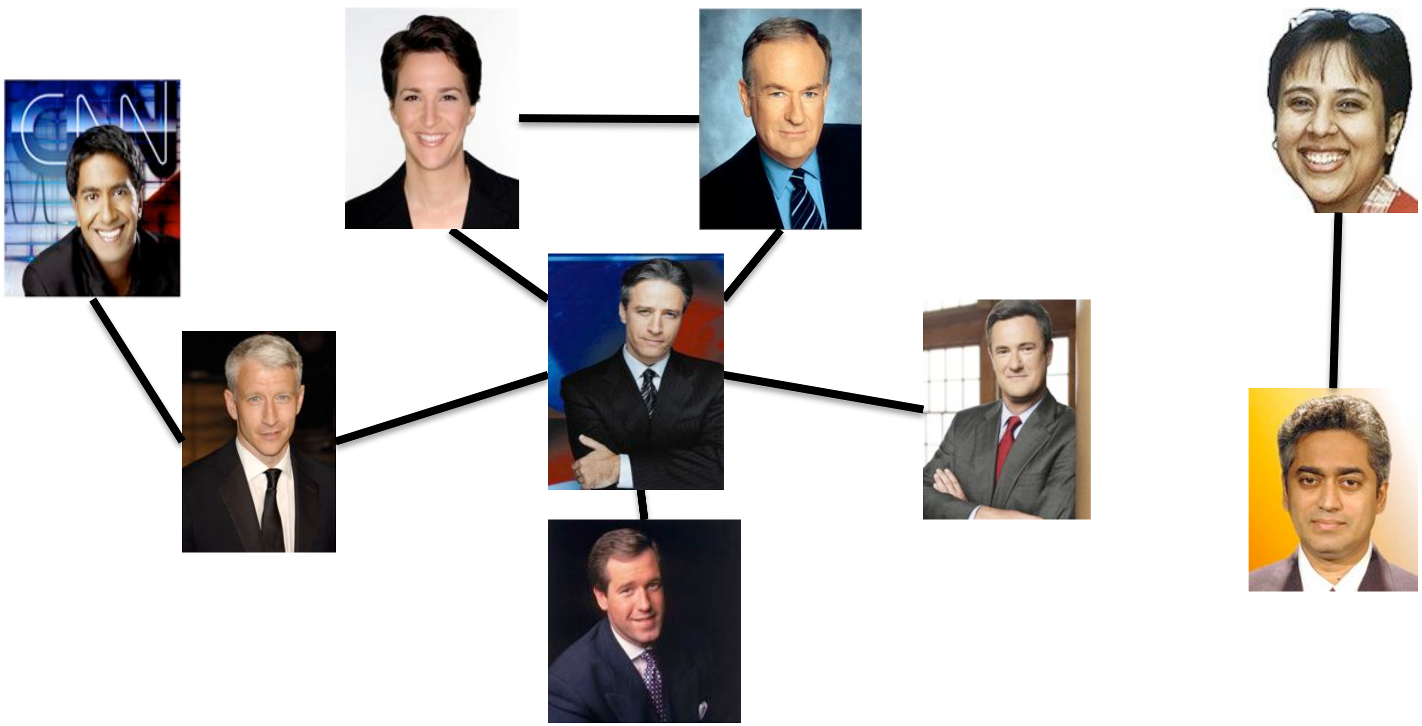


Solutions to HW 2

Handed out at the end of the lecture

Tree

Connected undirected graph with no cycles



Today's agenda

Prove that n vertex tree has n-1 edges

Algorithms for checking connectivity

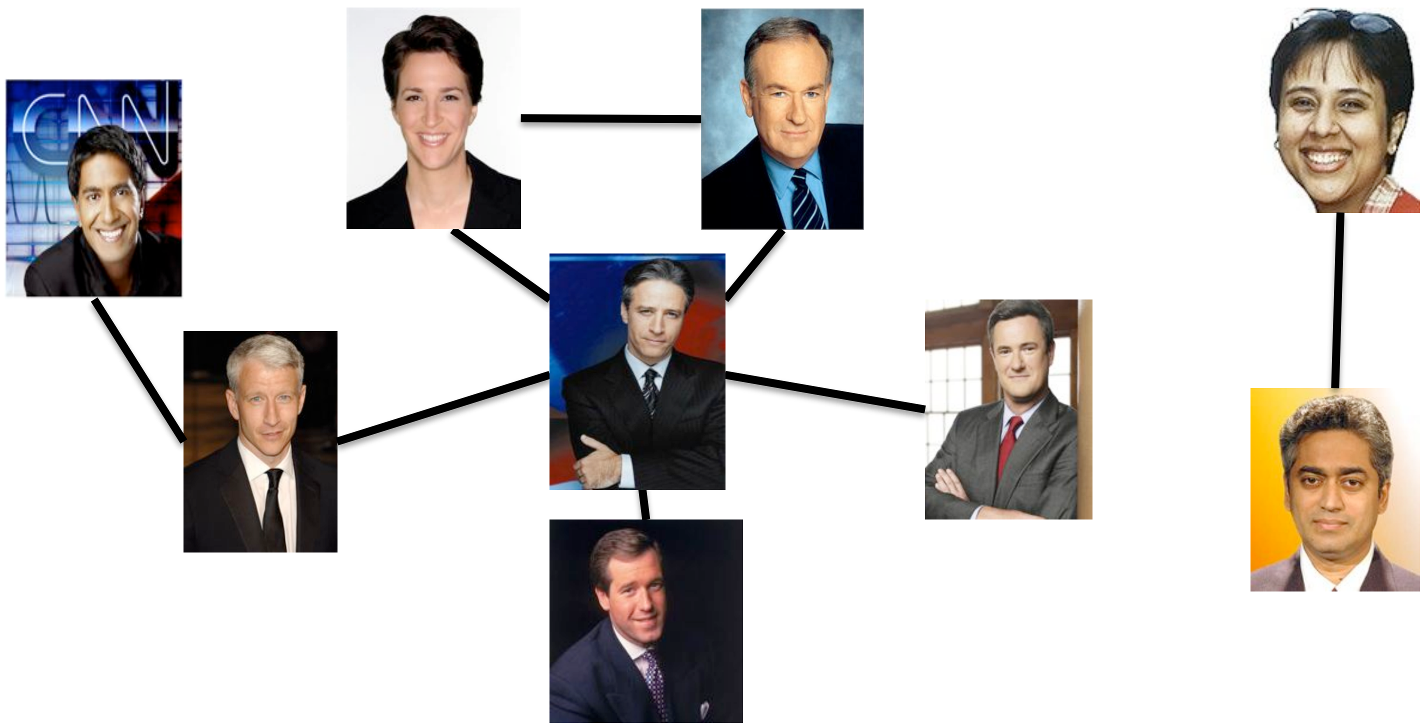
Questions?

Rest of Today's agenda

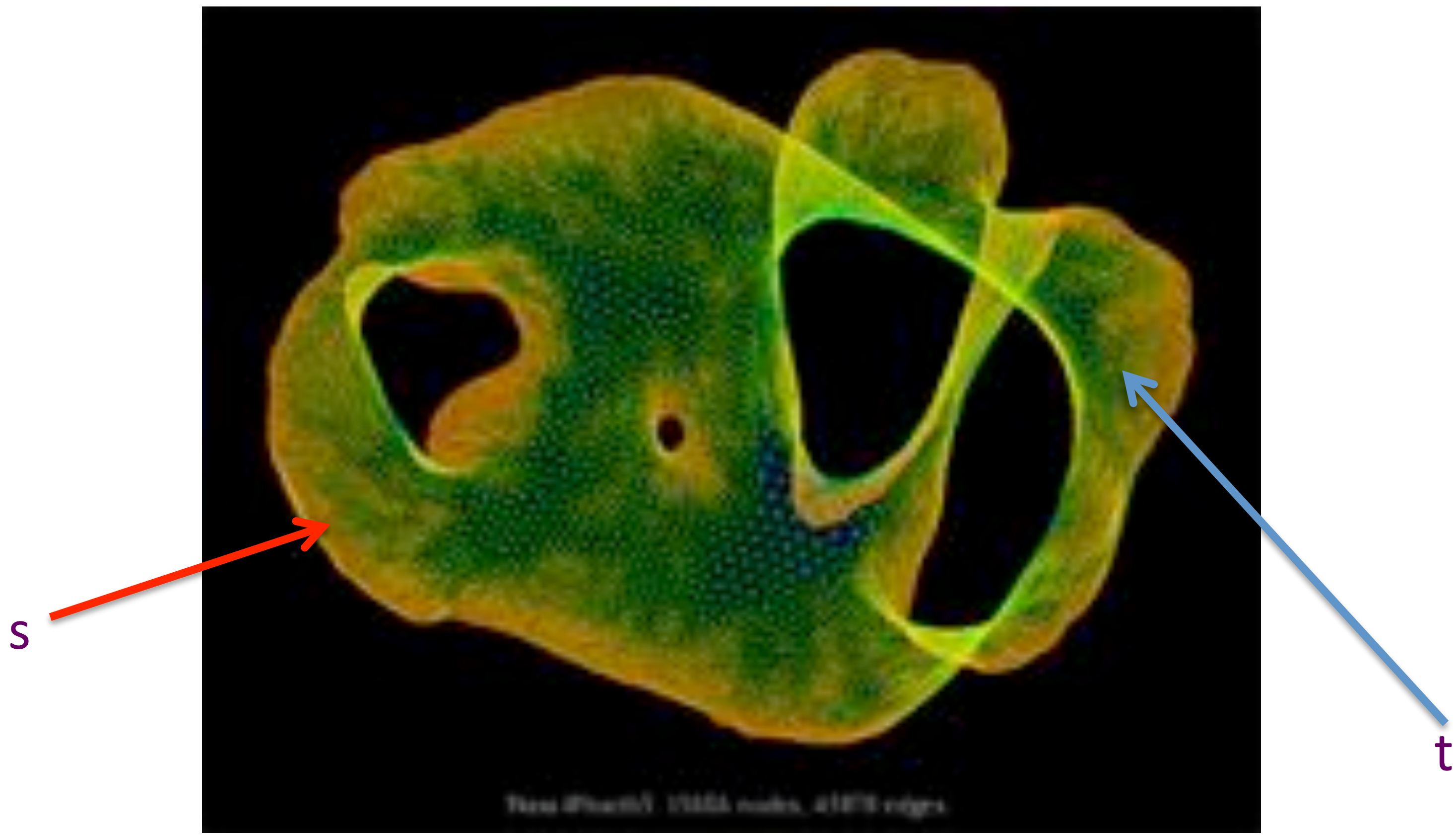
Finish Proving n vertex tree has n-1 edges

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

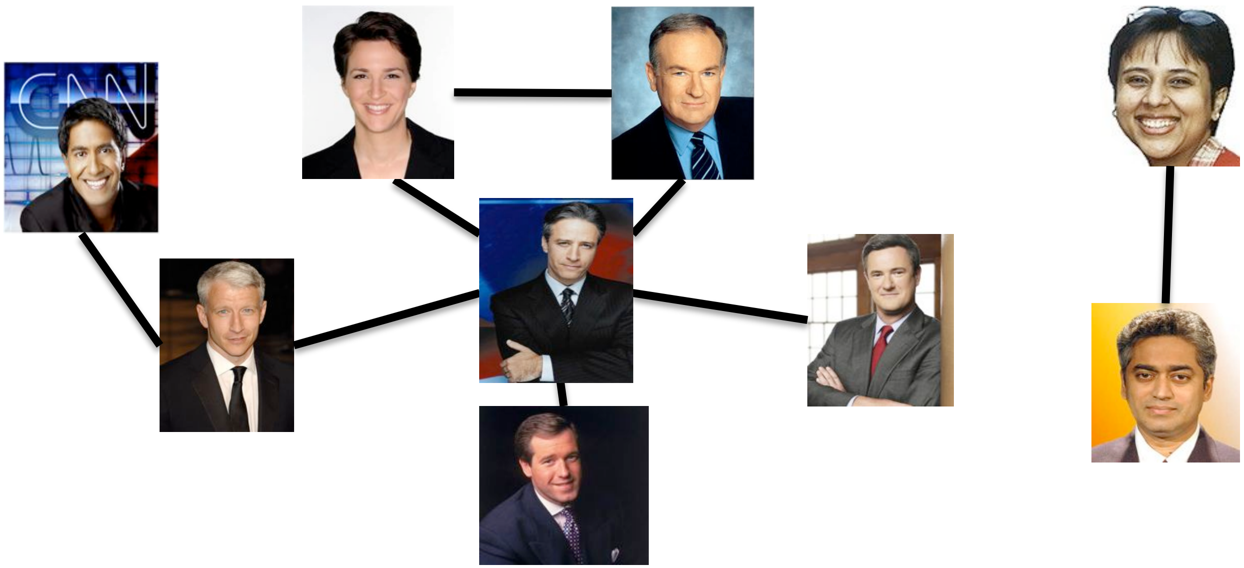
n^n such sequences

Check if any is a path between s and t

Algorithm motivation

Distance between u and v

Length of the shortest length path between u and v



Distance between RM and BO? 1

Questions?

Breadth First Search (BFS)

Is s connected to t?

Build layers of vertices connected to s

L_j : all nodes at distance j from 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

Exercise for you

Prove that L_j has all nodes at distance j from s

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”