Lecture 11

CSE 331

Sep 21, 2018

Mini Project group due Monday!

note

0 views

Actions

Clarifications on mini project

Two comments as you finalize your mini project choices:

- You are responsible for forming a group of size EXACTLY 3. I will not be forming groups from students who could not find a group at the end. **If you have not formed a group by Monday, then you get a zero on the mini project**.
- Once you submit a case study and it is not flagged as not having a conflict, your choice is considered to be final.
    - I'm open to considering requests for change but you need a good reason and this has to be done by email. I.e. do not fill in the form again-- if you do, I will simply delete your later choices.

mini_project

edit · good note | 0

Updated Just now by Atri Rudra

HW 3 is out!

Homework 3

Due by **11:59pm, Thursday, September 27, 2018**.

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

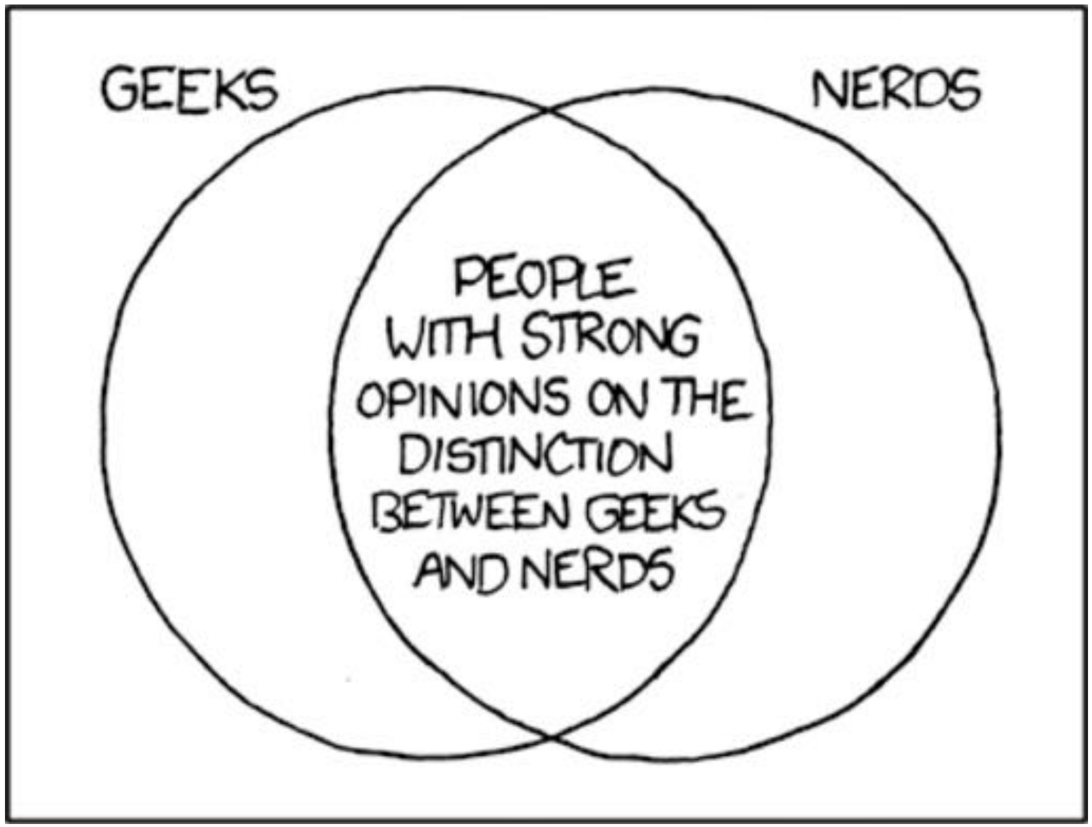
In this note we will be working with matrices and vectors. Simply put, matrices are two dimensional arrays and vectors are one dimensional arrays (or the "usual" notion of arrays). We will be using notation that is consistent with array notation. So e.g. a matrix $\mathbf{A}$ with m rows and n columns (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 $\mathbf{x}$ 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 $\mathbf{A}$ corresponding to the ith row and jth column as $A[i][j]$ (or `A[i][j]`). Similarly, the ith entry in the vector $\mathbf{x}$ 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):

Solutions to HW 2

Handed out at the end of the lecture

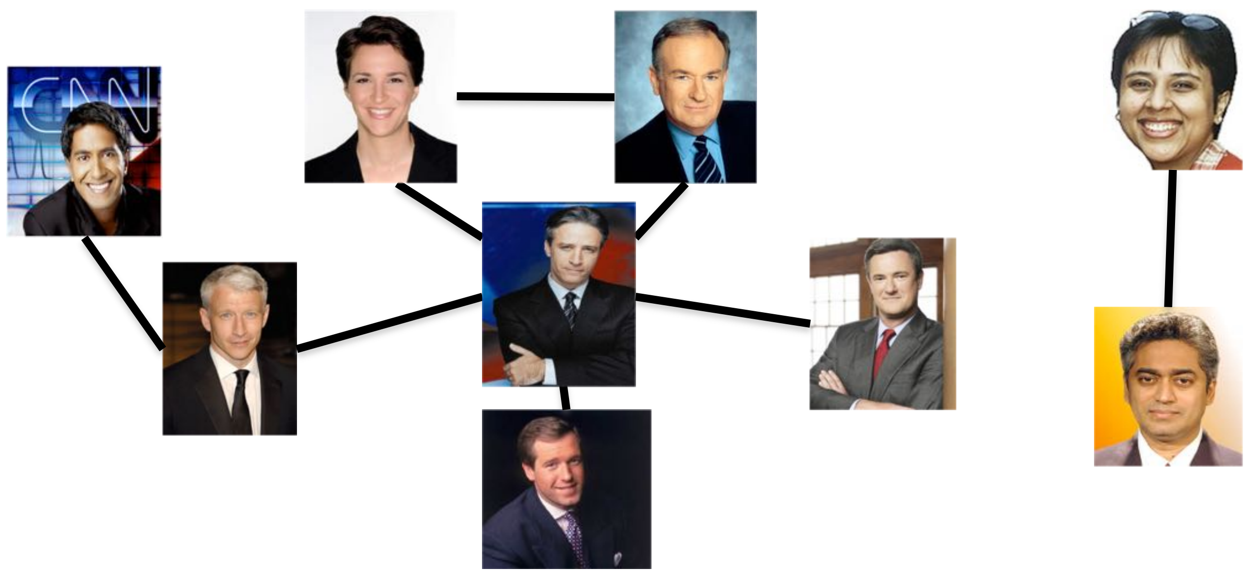
Formally define everything



http://imgs.xkcd.com/comics/geeks_and_nerds.png

Distance between u and v

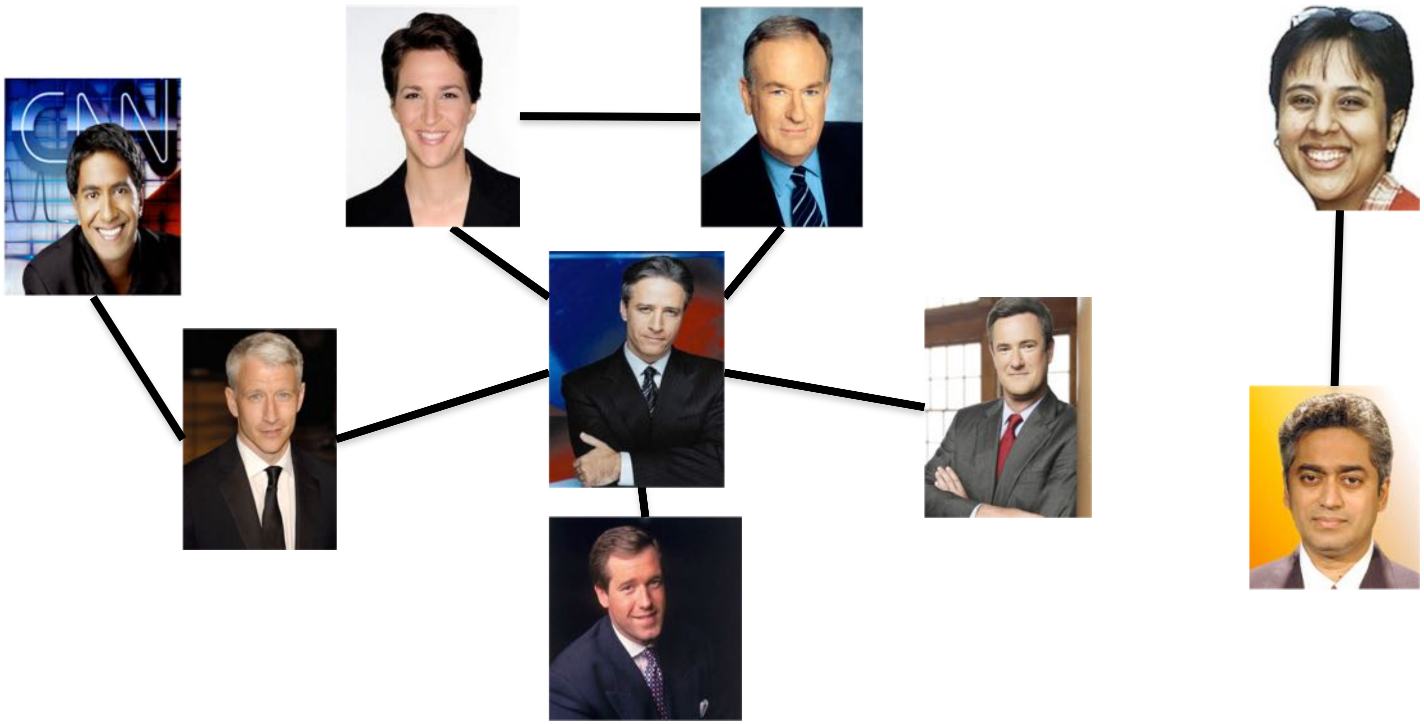
Length of the shortest length path between u and v



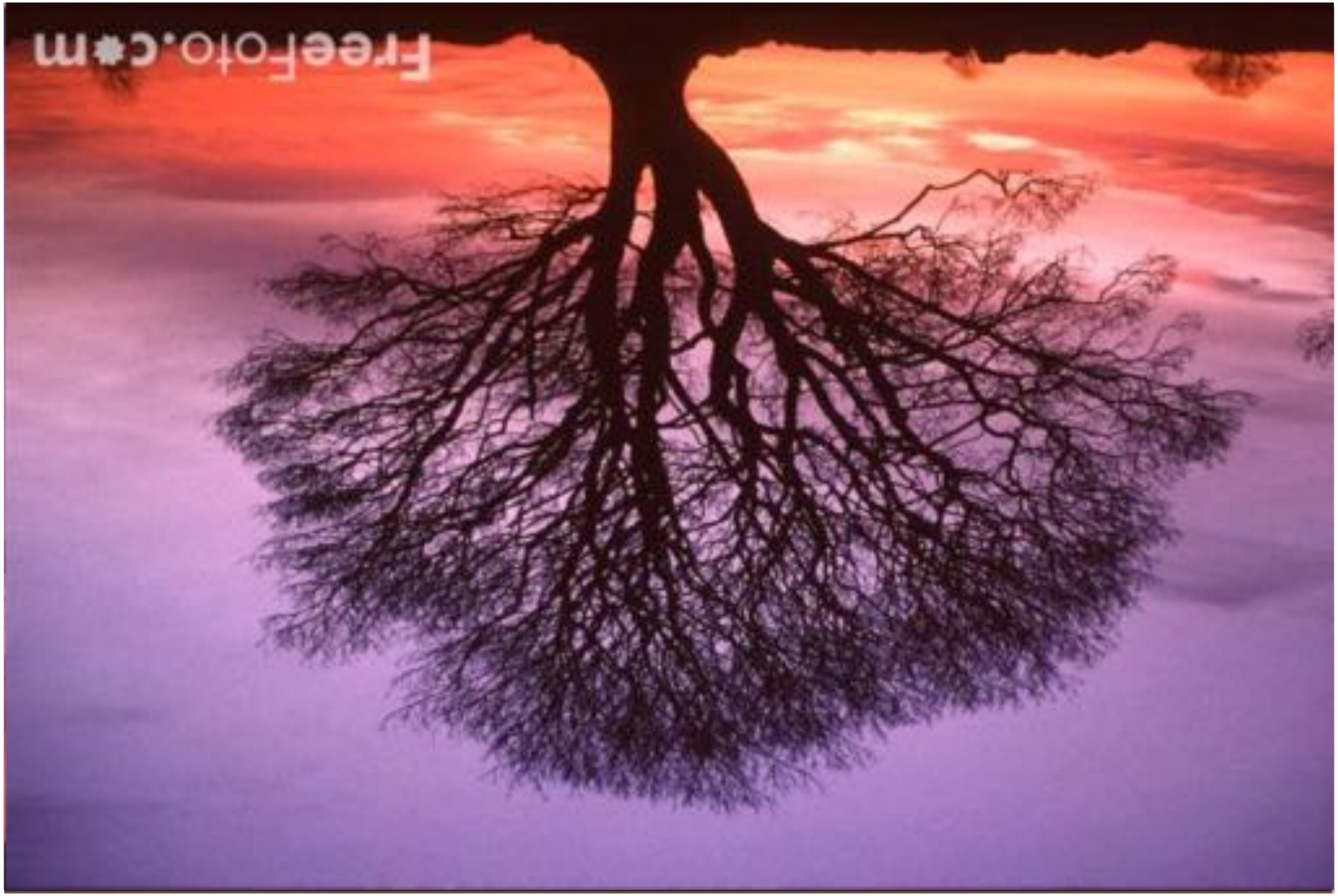
Distance between RM and BO? 1

Tree

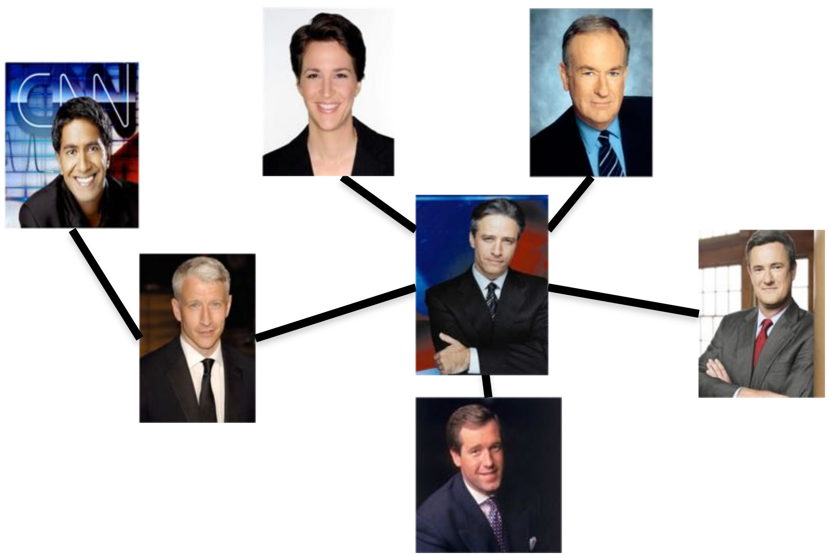
Connected undirected graph with no cycles

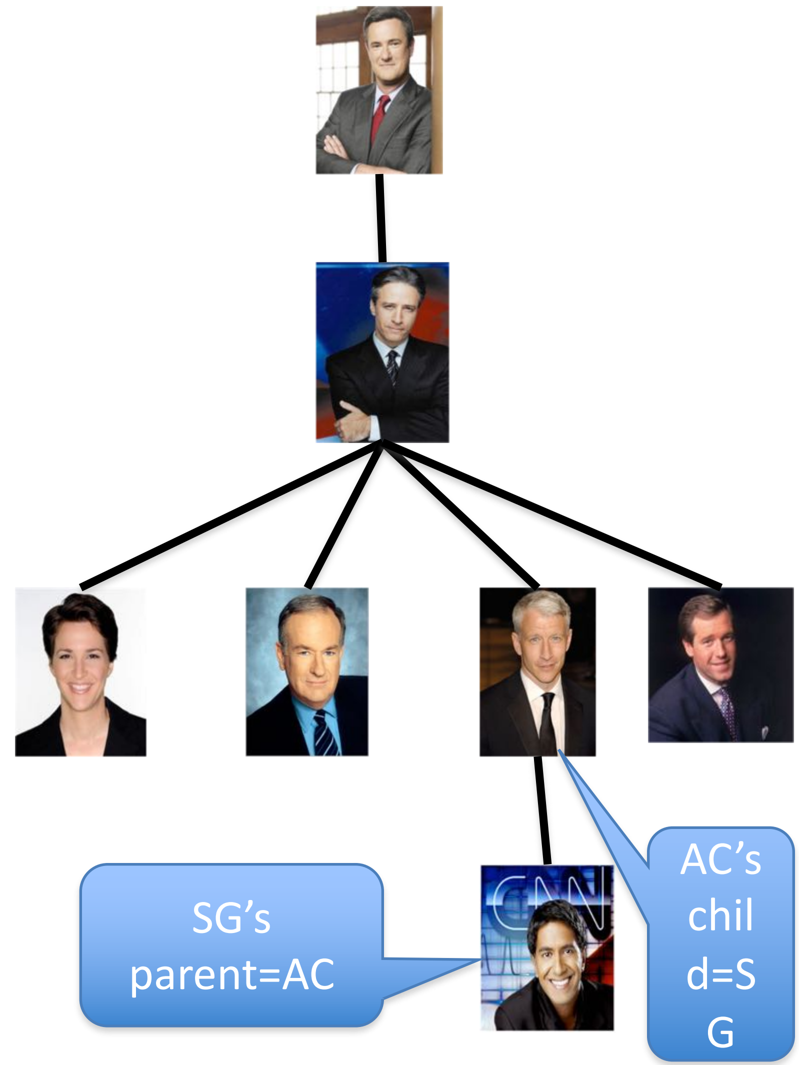


Rooted Tree



A rooted tree





Pick any vertex as root

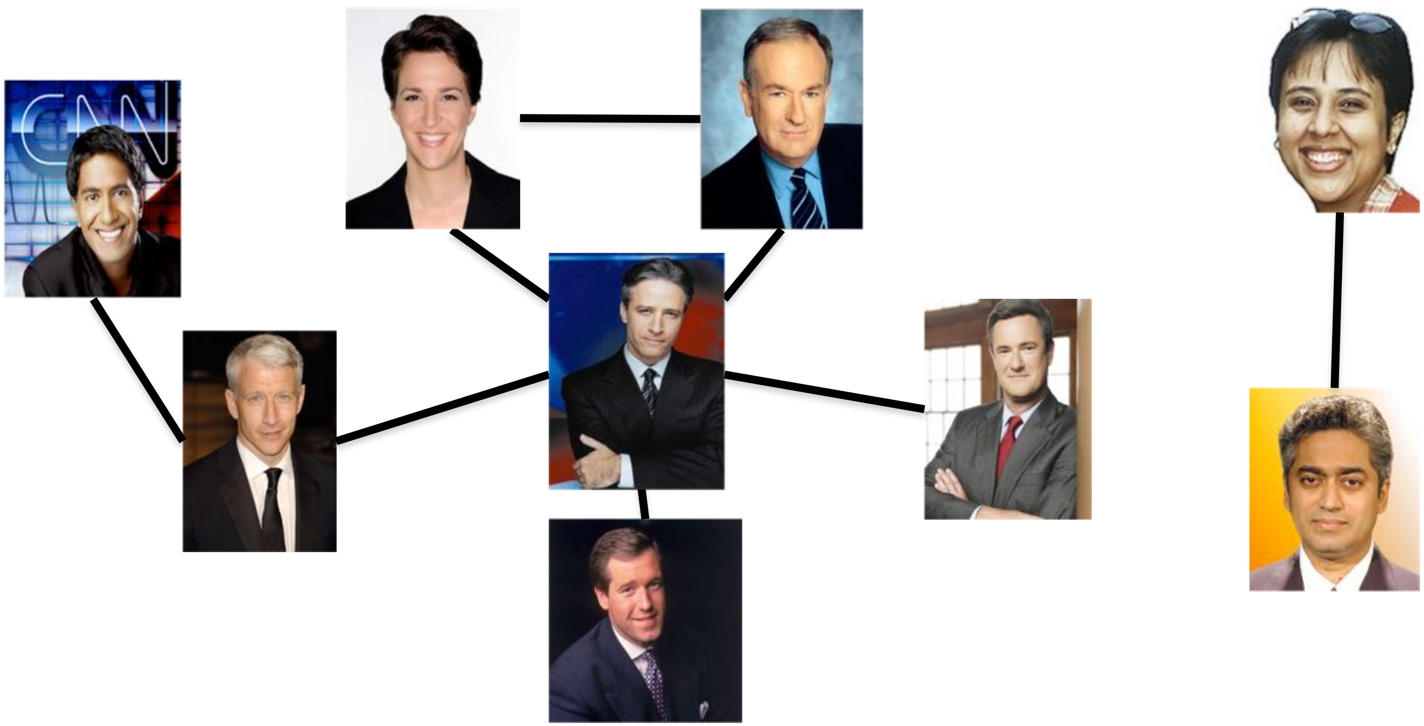
Let the rest of the tree hang under "gravity"

Rest of Today's agenda

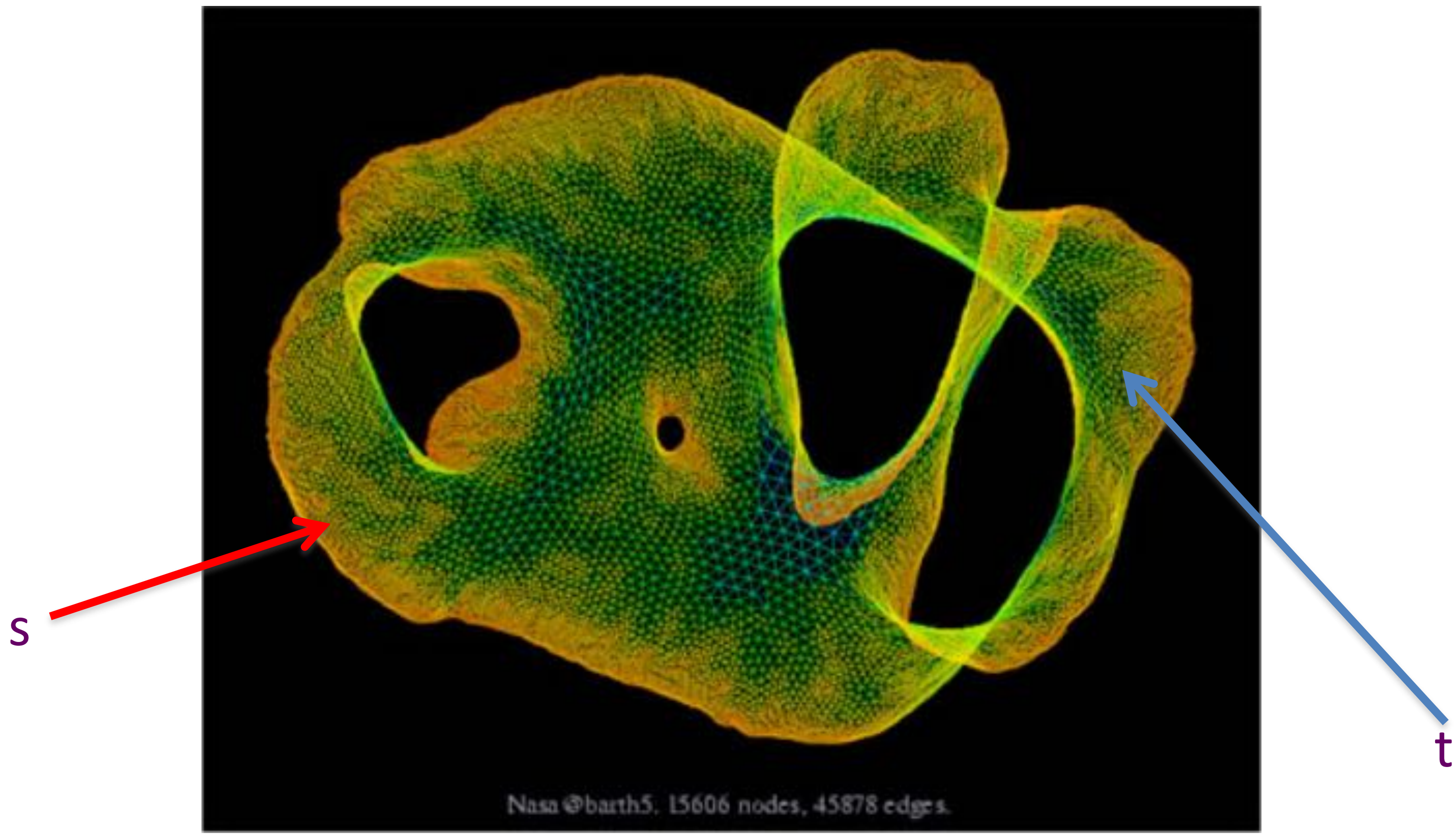
Prove n vertex tree has n-1 edges

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