Lecture 11

CSE 331

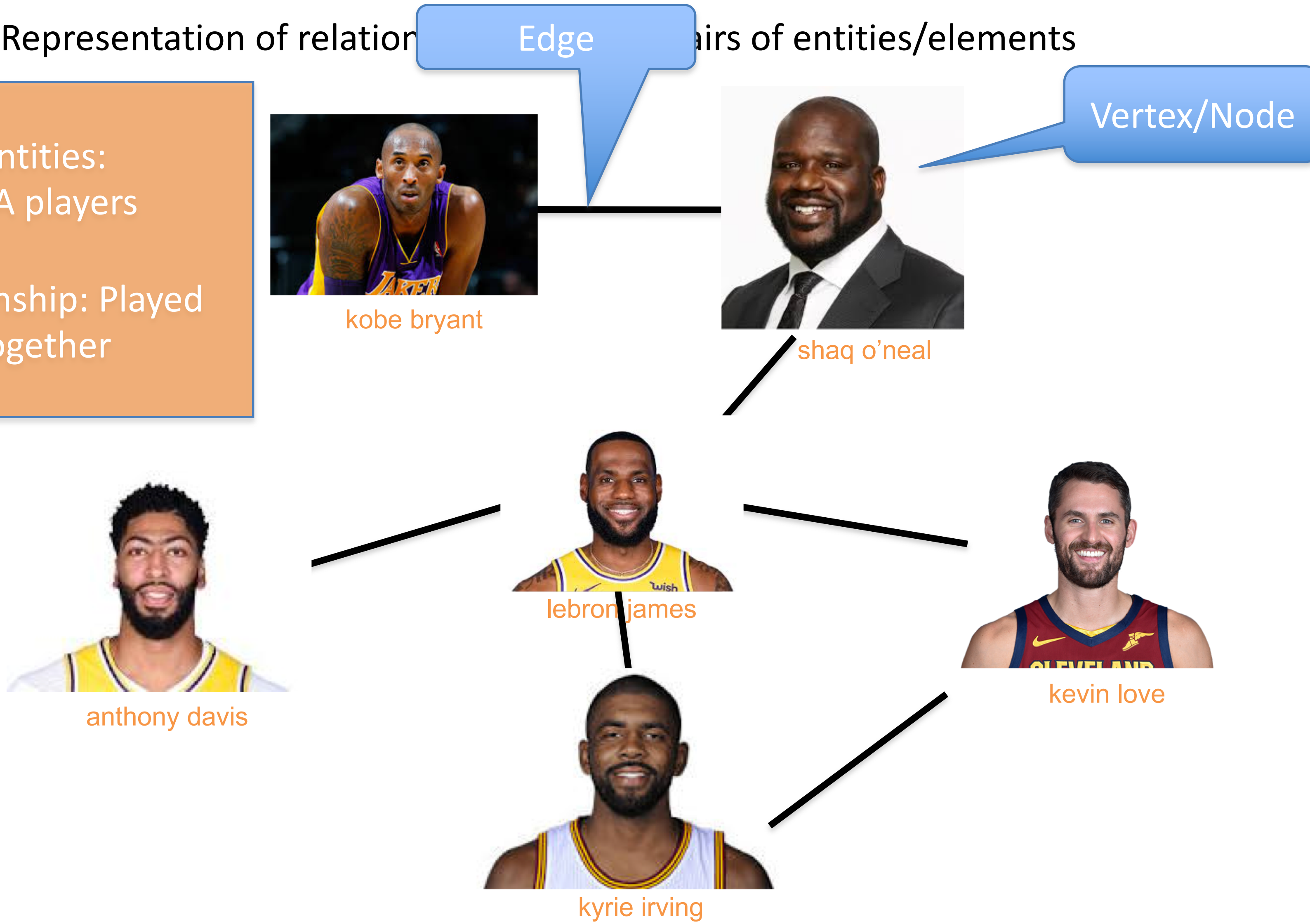
Feb 24, 2021

Graphs

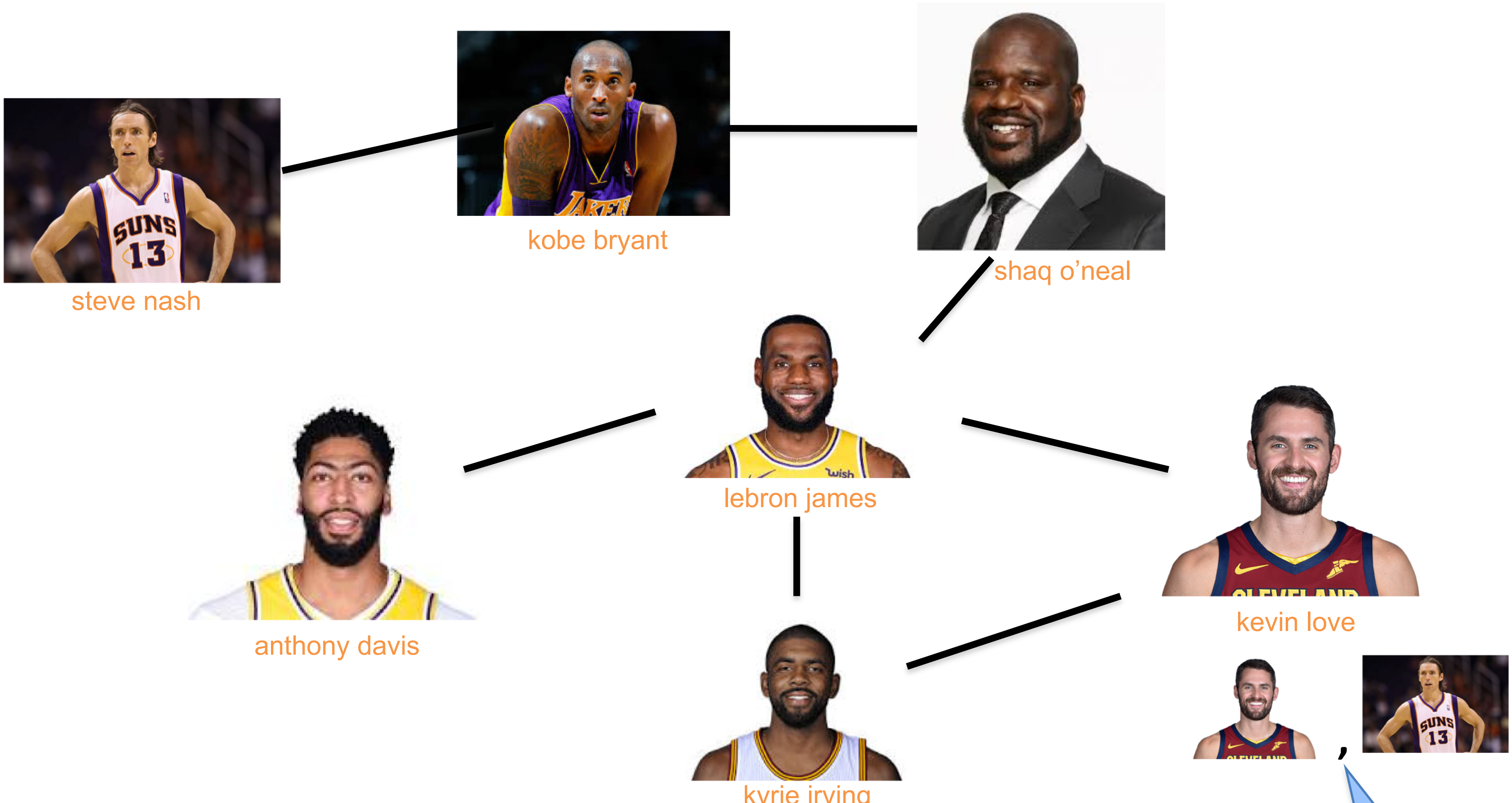
Representation of relation airs of entities/elements

Entities:
NBA players

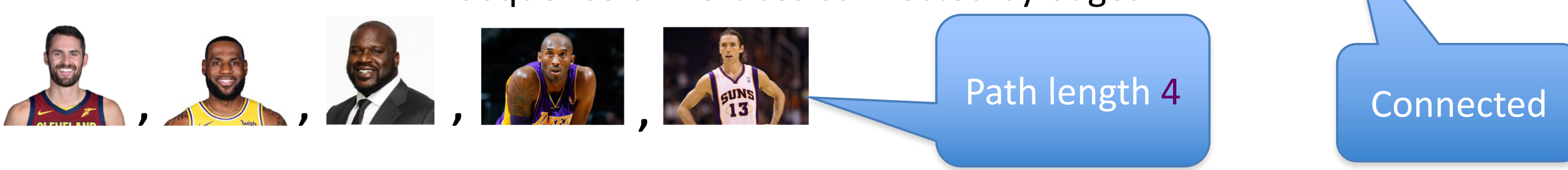
Relationship: Played together



Paths



Sequence of vertices connected by edges

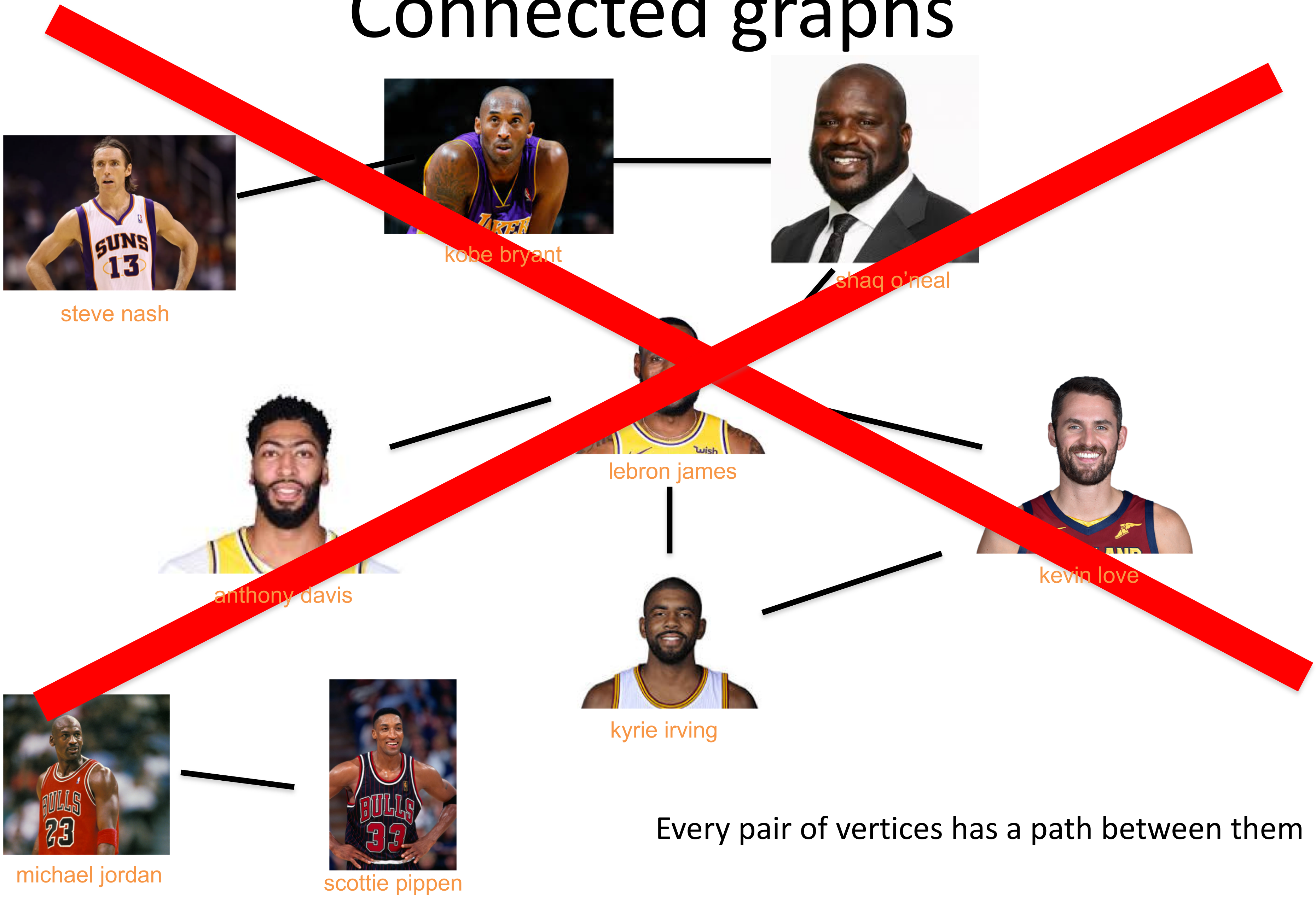


Connectivity

u and w are connected iff there is a path between them

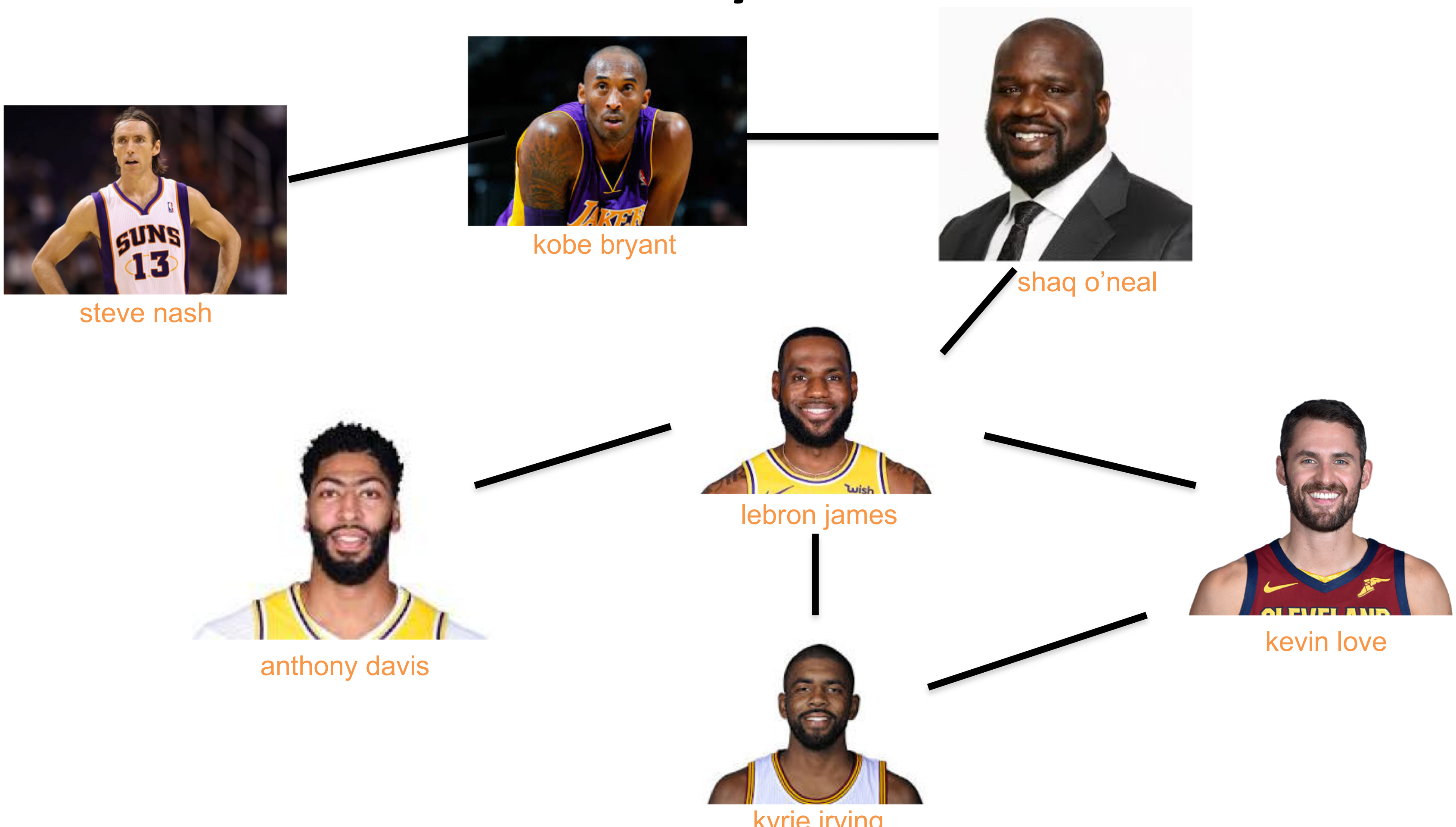
A graph is connected iff all pairs of vertices are connected

Connected graphs



Every pair of vertices has a path between them

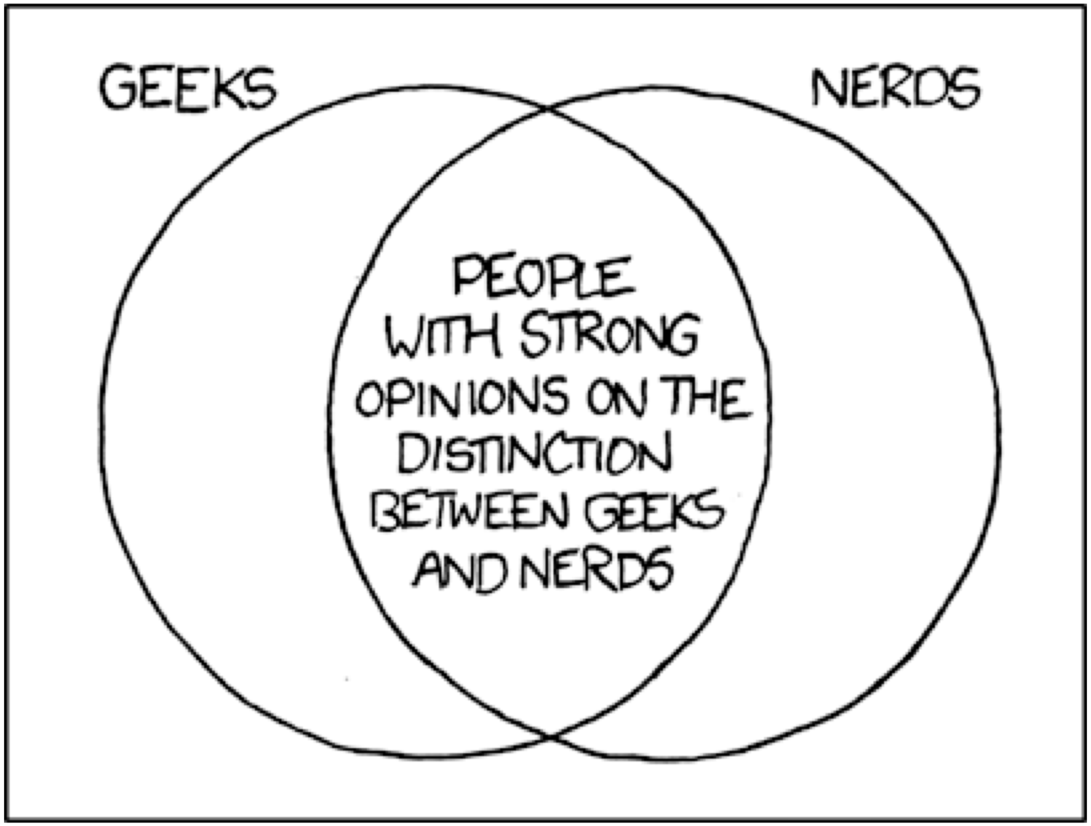
Cycles



Sequence of k vertices connected by edges, first k-1 are distinct

Questions?

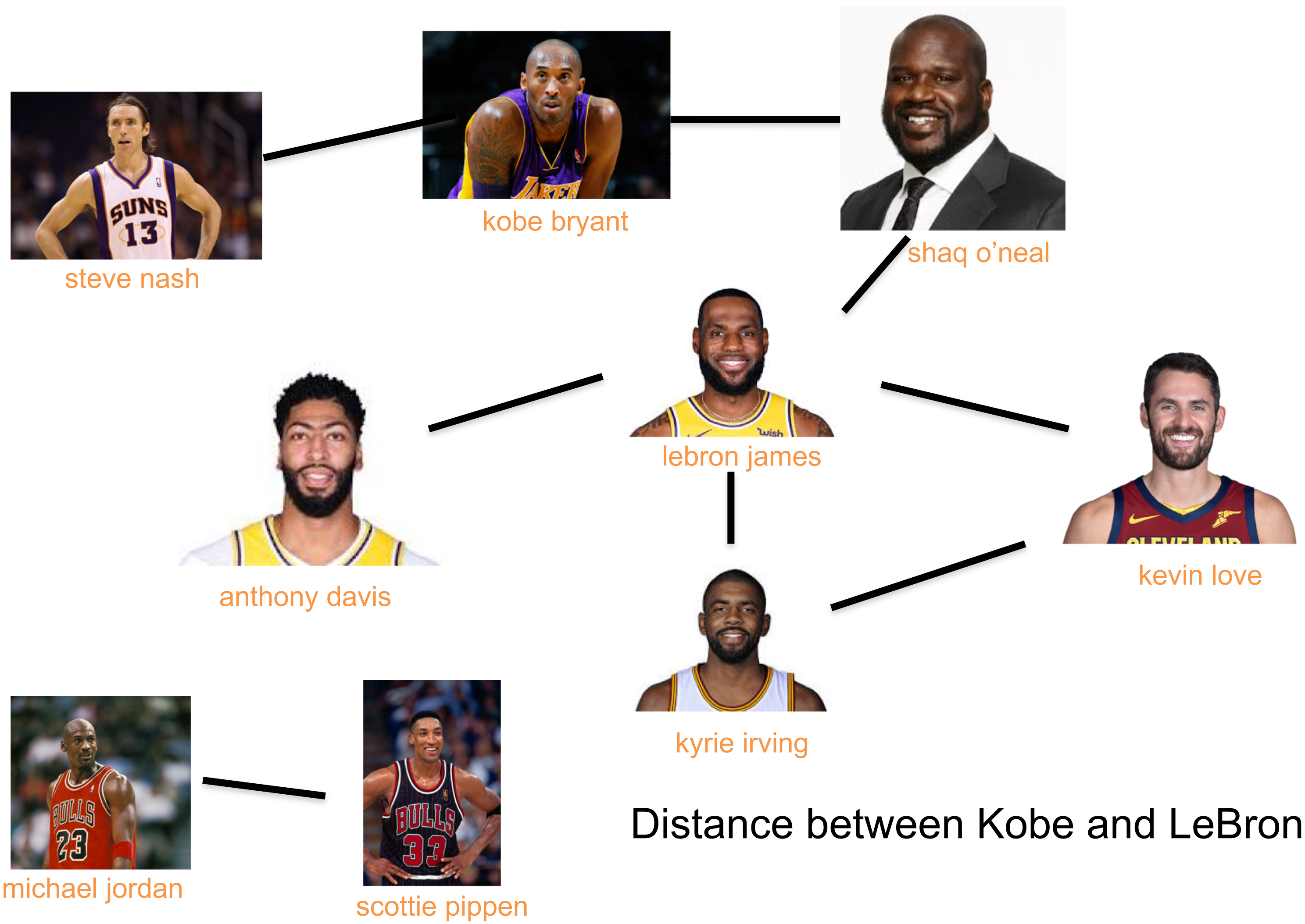
Formally define everything



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

Distance between u and v

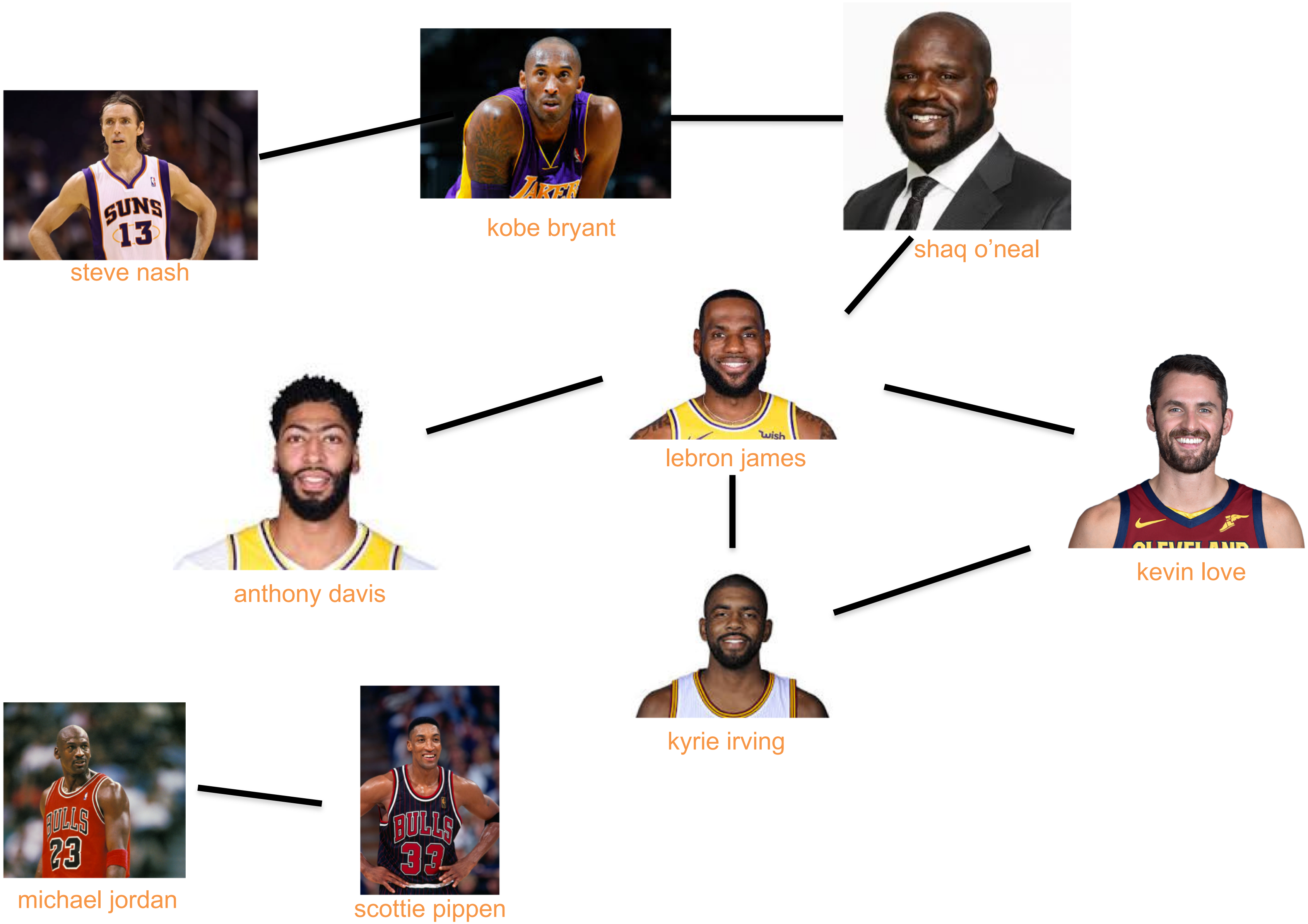
Length of the shortest length path between u and v



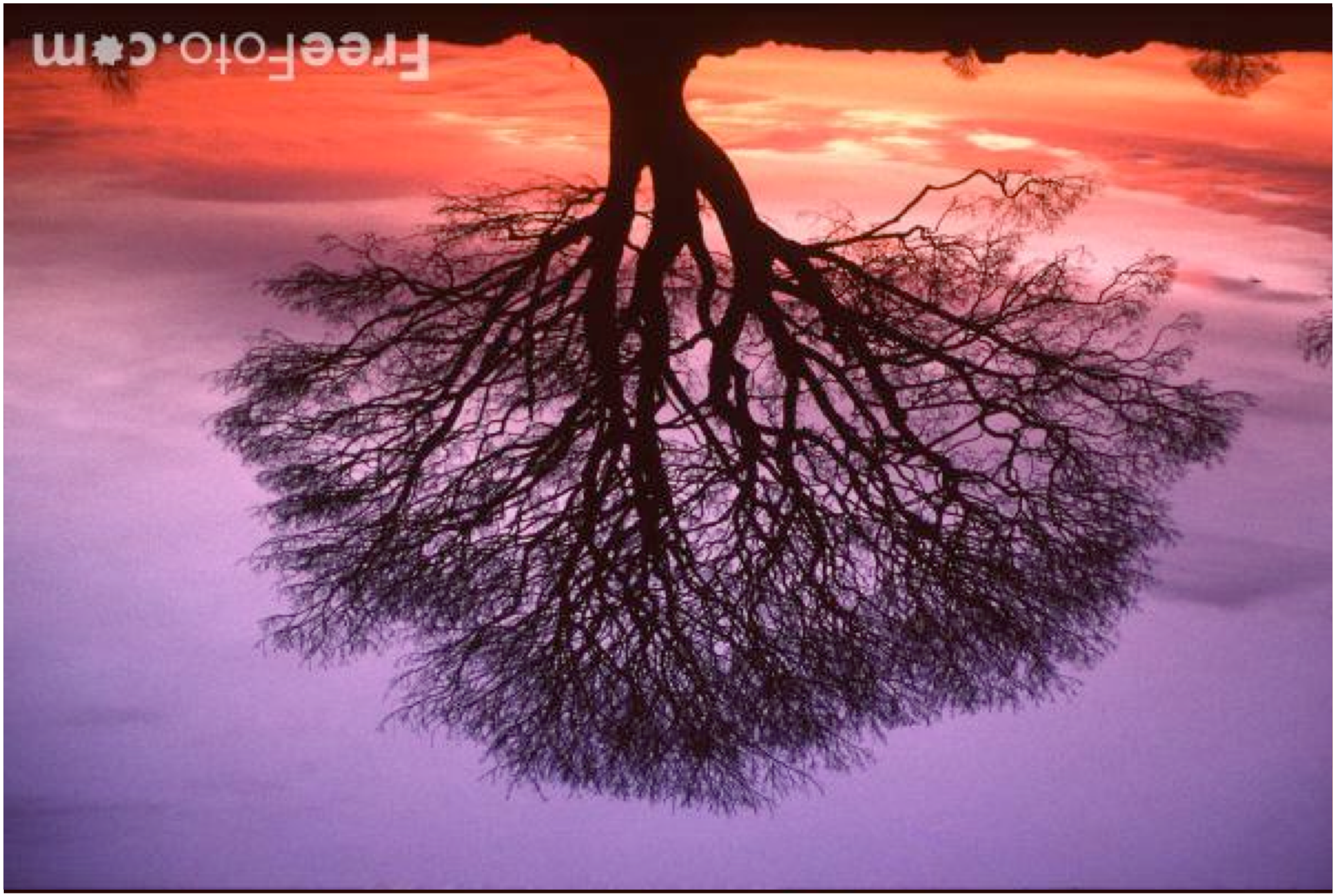
Distance between Kobe and LeBron? 2

Tree

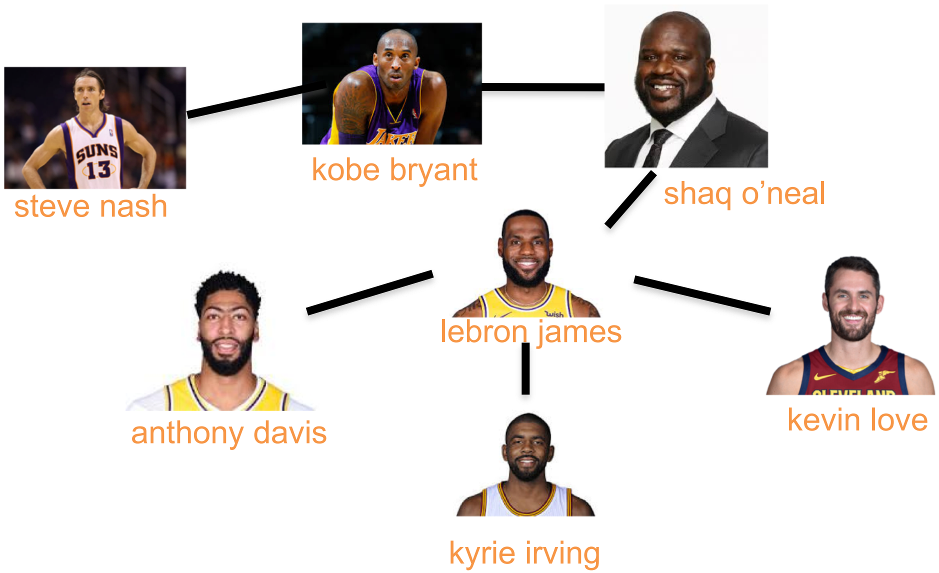
Connected undirected graph with no cycles

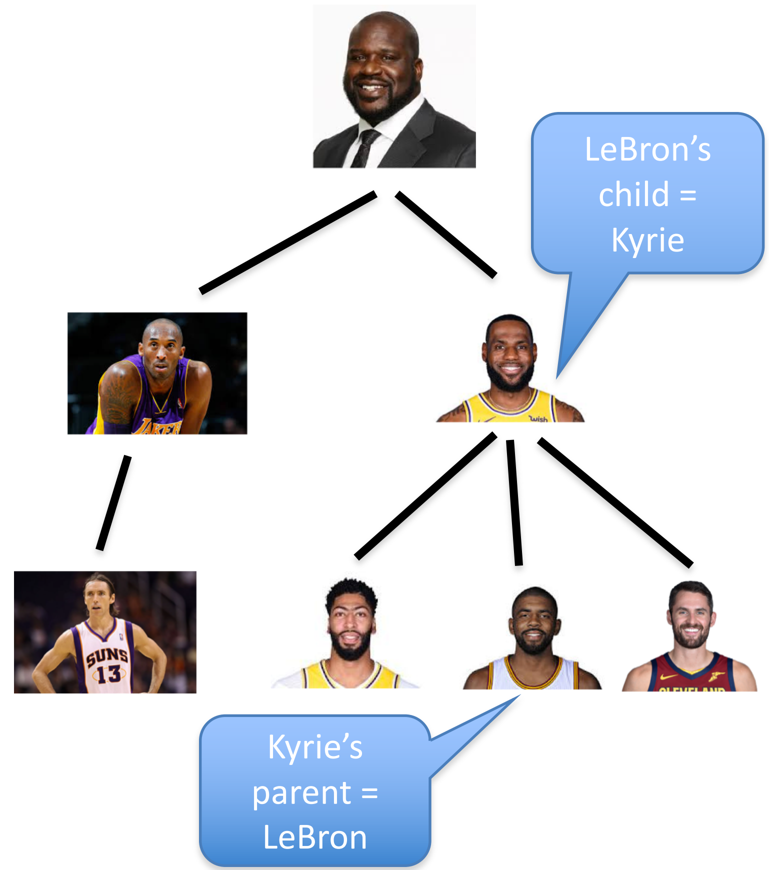


Rooted Tree



A rooted tree





Pick any vertex as root

Let the rest of the tree hang under "gravity"

Every n vertex tree has n-1 edges

Trees

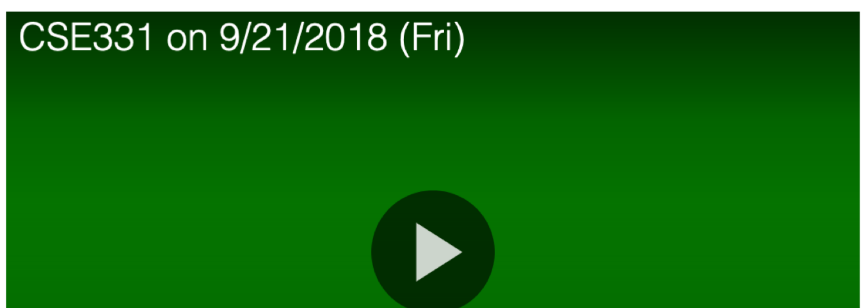
This page collects material from previous incarnations of CSE 331 on trees, especially the proof that trees with n nodes have exactly $n-1$ edges.

Where does the textbook talk about this?

`Section 3.1` in the textbook has the lowdown on trees.

Fall 2018 material

Here is the lecture video:



Every n vertex tree has n-1 edges

Let G be an undirected graph on n nodes

Then ANY two of the following implies the third:

T is connected

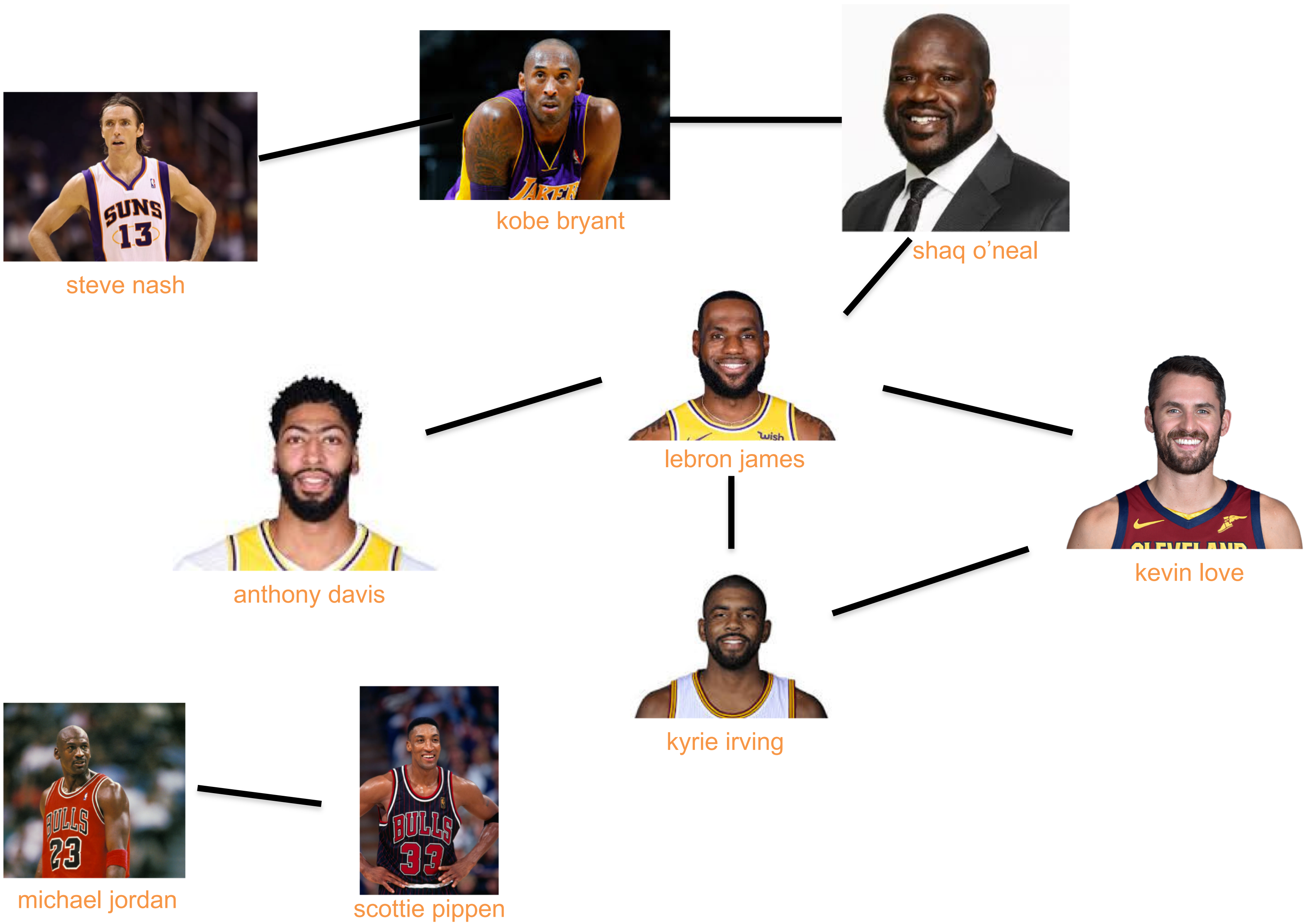
T has no cycles

T has n-1 edges

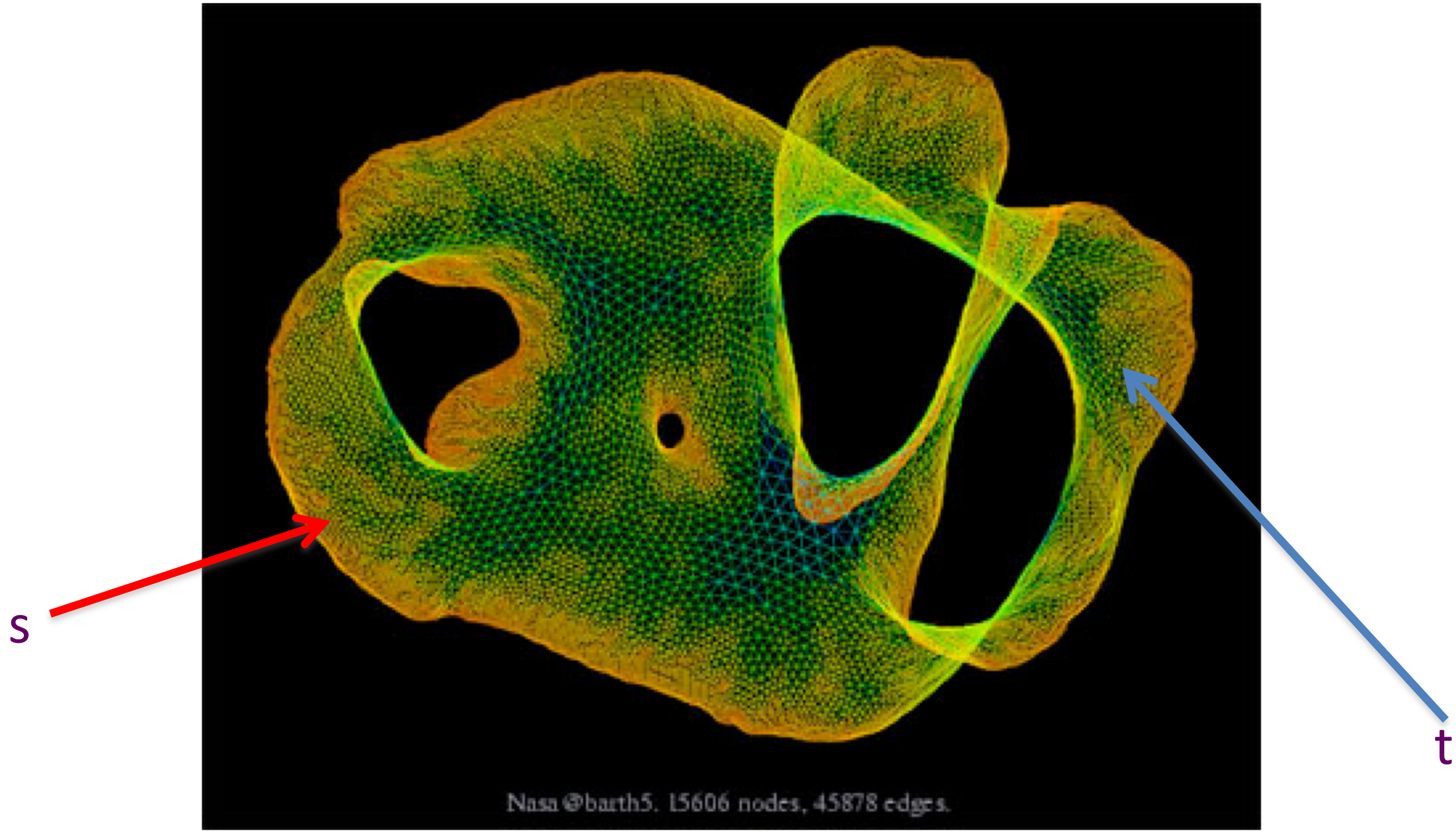
Rest of Today’s agenda

Algorithms for checking connectivity

Checking by inspection



What about large graphs?



Are s and t connected?

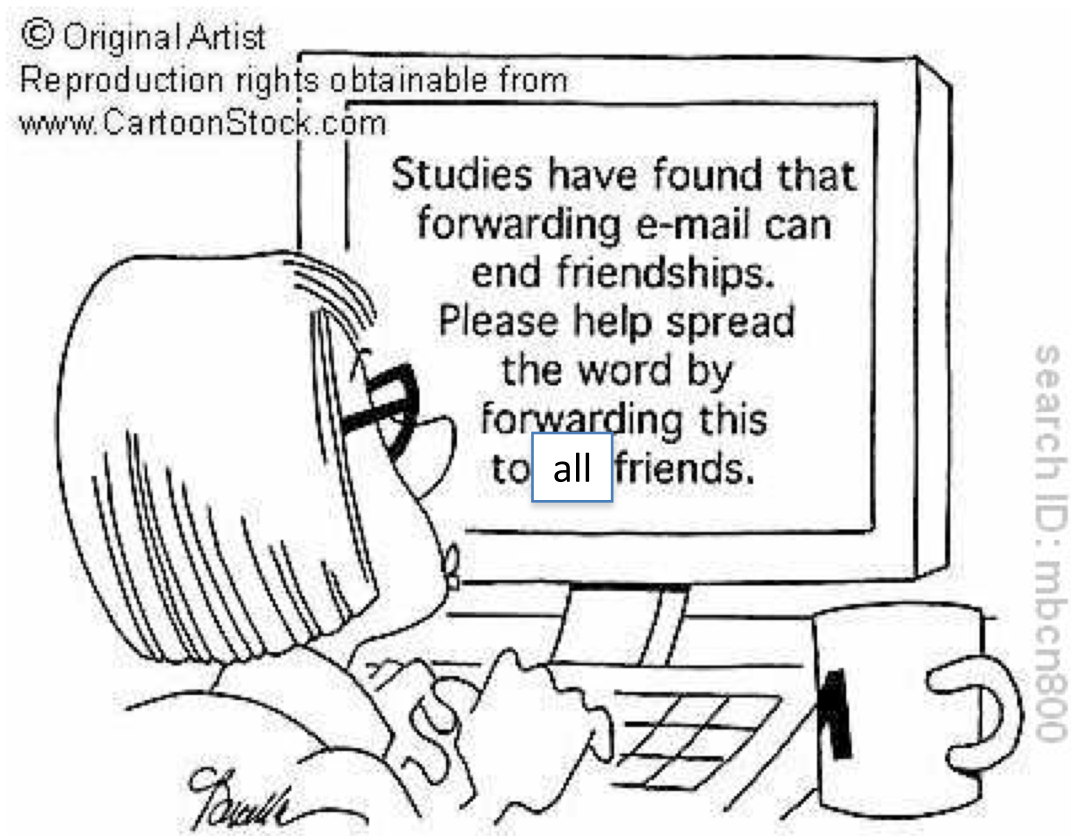
Brute-force algorithm?

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



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:

Connectivity Problem

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

Output: All t connected to s in G

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} is the set of vertices not chosen yet but are connected by an edge to L_j

Stop when new layer is empty