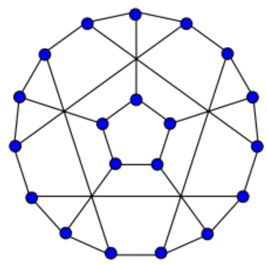
Ma/CS 6a

Class 6: Introduction to Graphs



By Adam Sheffer

Today's Class is about

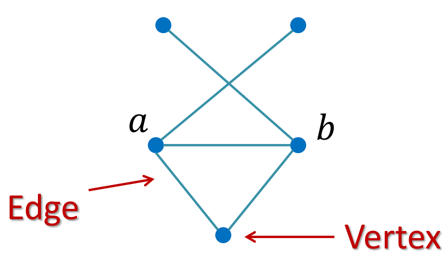
Six Degrees of Kevin Bacon

Six Degrees of Kevin Bacon

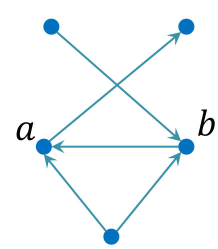
- **Claim.** Any actor can be linked through his/her film roles to Kevin Bacon within six steps.
- **Example.** Keanu Reeves:

Graphs

Undirected graph



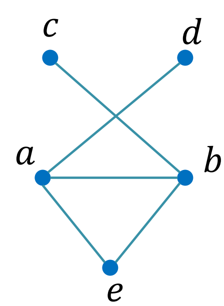
Directed graph



Graph Representation

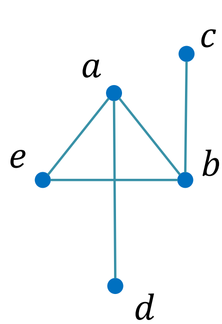
- We write $G = (V, E)$. That is, the graph G has vertex set V and edge set E.

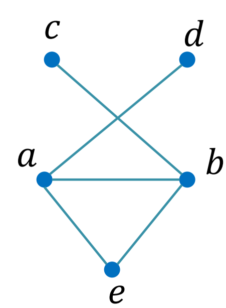
- **Example.** In the figure:
  - $V = \{a, b, c, d, e\}$.
  - $E = \{(a, b), (a, d), (a, e),$ $(b, c), (b, e)\}$.



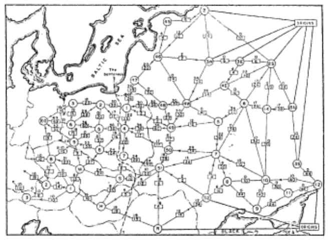
Graph Representation (cont.)

- $V = \{a, b, c, d, e\}$.
- $E = \{(a, b), (a, d), (a, e), (b, c), (b, e)\}$.

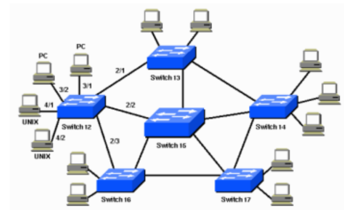




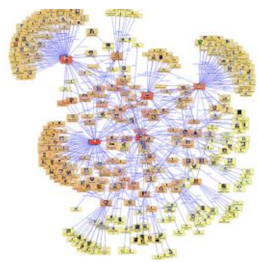
What is this Good For?



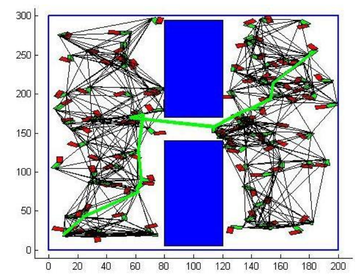
Cold war analysis



Communication networks

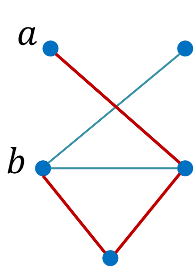


Social networks

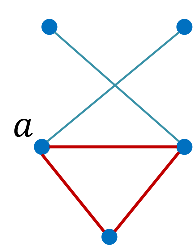


Robot Motion Planning

Paths and Cycles



Path between a and b.

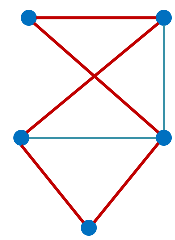


Cycle through a

A *cycle* is a path that starts and ends in the same vertex.

More on Paths and Cycles

- A path/cycle is said to be ***simple*** if it does not visit any vertex more than once.
- The length of a path/cycle is the number of edges that it consists of.
- **Example.** A simple cycle of length 5.

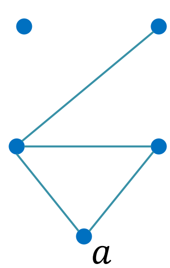


Connectedness and Degrees

- A graph is ***connected*** if there is a path between any two vertices.
- The ***degree*** of a vertex is the number of edges that are adjacent to it.

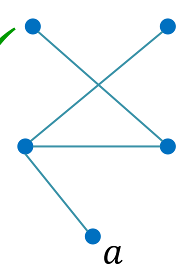
Connected? ✗

$\deg(a) = 2$



Connected? ✓

$\deg(a) = 1$

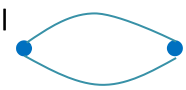


Simple Graphs

- An edge is a *loop* if both of its endpoints are the same vertex.
- Two edges are *parallel* if they are between the same pair of vertices.
- A graph is *simple* if it contains no loops and no parallel edges.
- For now, we only consider simple graphs.

A loop

Parallel edges

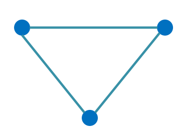


Warm-up Exercise

- **Prove.** In any graph, the sum of the degrees of the vertices is even.

- **Proof.** Every edge contributes 1 to the degree of exactly two vertices. Thus,

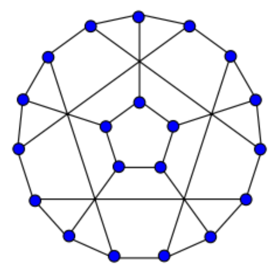
$$\sum_{v \in V} \deg(v) = \sum_{e \in E} 2 = 2|E|.$$



Paths and Degrees

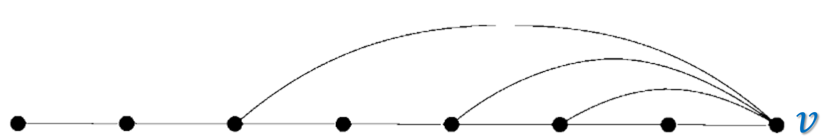
- **Problem.** Let $G = (V, E)$ be a graph such that the degree of every $v \in V$ is at least d (for some $d \geq 2$). Prove that G contains a path of length d.

A graph with minimum degree 3.



Proof

- Assume, for contradiction, that a longest path P is of length $c < d$.
- Consider a vertex v which is an endpoint of P.
- Since $\deg v \geq d \geq c + 1$, it must be connected to at least one vertex $u \notin P$.
- By adding the edge (v, u) to P, we obtain a longer path, contradicting the maximality of P.



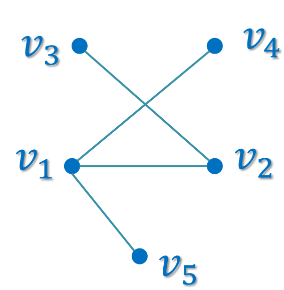
A Variant of the Problem

- **Problem.** Let $G = (V, E)$ be a graph such that the degree of every $v \in V$ is at least d (for some $d \geq 2$).
  - What is the minimum length of a cycle in G?

$$d + 1$$

Connectivity Problem

- **Prove.** The vertices of a connected graph G can always be ordered as $\{v_1, v_2, \ldots, v_n\}$ such that for every $1 < i \leq n$, if we remove $\{v_i, v_{i+1}, \ldots, v_n\}$ and the edges adjacent to these vertices, G remains connected.



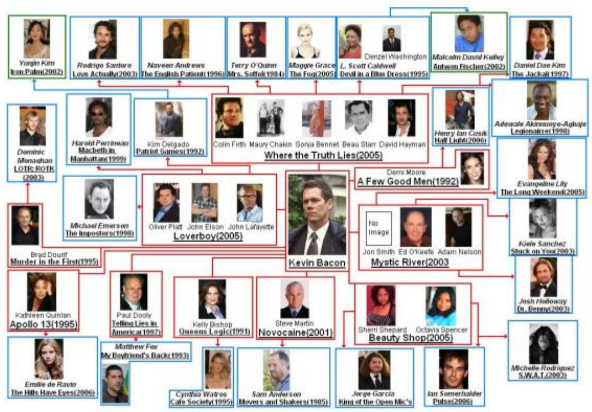
Proof

- Pick any vertex as v_1.
- Pick a vertex that is connected to v_1 in G and set it as v_2.
- Pick a vertex that is connected either to v_1 or to v_2 in G and set it as v_3.
- …

Back to Bacon

- We wish to build a graph for the problem.
- What are the vertices of the graph?
  - A vertex for each actor.
- When is there an edge between two vertices?
  - When the corresponding actors played in a common movie.
- Is the graph directed?
  - No.

Example



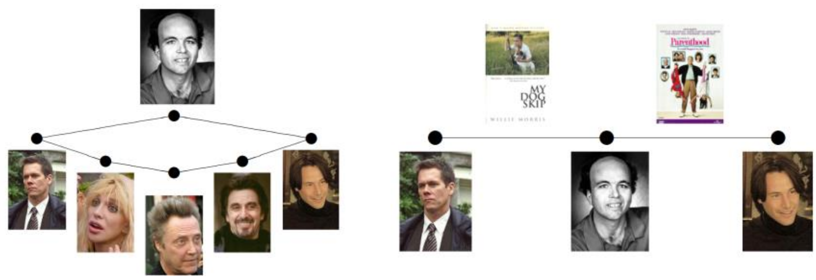
- We want to check if every actor has a finite Bacon Number. What should we check?
  - Whether the graph is connected.

Bacon Numbers

- The *Bacon number* of an actor is the minimum number of steps required to connect him to Kevin Bacon.
- Example. By the picture below:
  - Christopher Walken’s Bacon number is 2.
  - Keanu Reeves’ Bacon number is 4.

Bacon Numbers (2)

- But Keanu Reeves' Bacon number is 2!



- What should we do in the graph to find the correct Bacon number of an actor?
  - The length of the shortest path from the actor's vertex to Bacon's vertex.

The BFS Algorithm

- We wish to find out:
  - Whether the graph is connected.
  - The shortest paths from Bacon's vertex to every other vertex.
- *The BFS algorithm*:
  - **Input.** An undirected graph $G = (V, E)$ and a vertex $s \in V$.
  - **Output.** A shortest path from s to any other vertex of G (if such a path exists).
  - G is connected if and only if there is a path to every vertex.

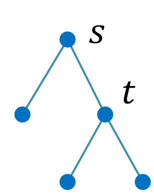
Trees and Forests

- In an undirected graph, a ***tree*** is a connected subgraph containing no circles.
- A ***forest*** is a set of non-connected trees.

A forest with four trees

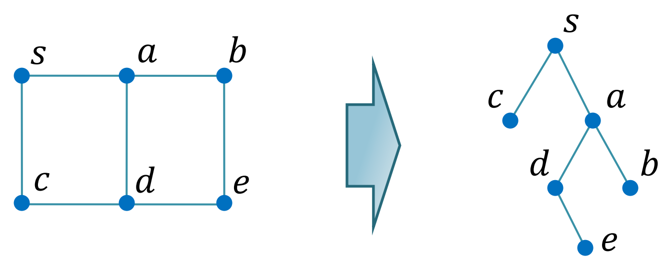
Rooted Trees

- A ***rooted tree*** is a tree with a special vertex – the ***root*** – that is singled out.
- We draw the tree with the root on top, and the edges "grow downwards".
- A vertex v is the ***parent*** of a vertex u if there is an edge (u, v) and v is above u.
  - Each vertex, except for the root, has a unique parent.



s is the root and t's parent

BFS Output

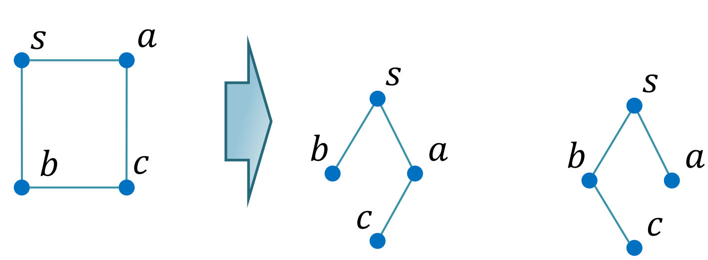


- The output is a ***BFS tree***, containing only shortest paths from s.
  - A rooted tree with root s.

Test Your Intuition

- **Problem.** Is there always a unique tree containing the shortest paths from s?

- **Answer.** No!



Erdős Numbers: the Math Version

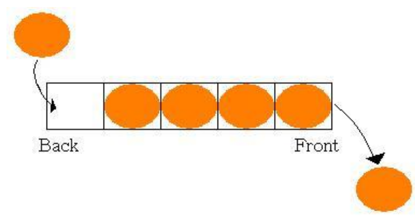
- **Paul Erdős (1913-1996).** A Hungarian mathematician. Possibly the most prolific mathematician ever.
  - People that wrote a paper with Erdős have an *Erdős number* of 1.
  - People that wrote a paper with someone that has an Erdős number of 1, have an Erdős number of 2.
  - Etc.
- Most leading scientists have a small Erdős number.

BFS: Colors

- We call the vertex s that we start from the *root* of the tree.
- BFS scans the graph starting from the root.
- During the scan, every vertex has a color:
  - Vertices that the algorithm did not visit yet are colored *white*.
  - Vertices that the algorithm did visit, but is not yet done with are colored *gray*.
  - Vertices that the algorithm is done with are colored *black*.

Data Structure: Queue

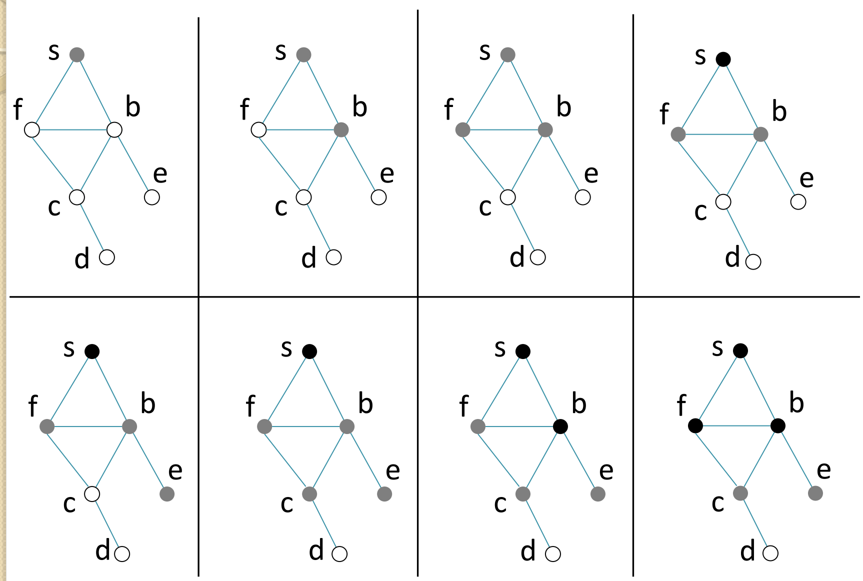
- A ***queue*** stores "objects" (in our case – vertices).
- Supports the operations:
  - Enqueue – insert an object to the back of the queue.
  - Dequeue – remove an object from the front of the queue.



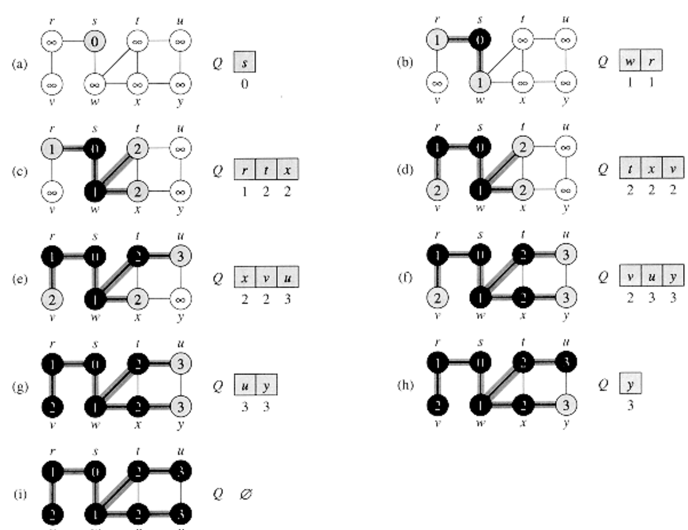
BFS: The Main Idea

- A queue Q holds the vertices that are currently gray. At first $Q = \{s\}$.
- At each step, take out a vertex $u \in Q$ and for every edge e adjacent to u:
  - If the other vertex of e is gray or black, do nothing.
  - If the other vertex of e is white, color it gray and insert it into Q.
- After going over all of u's edges, color u black, and move to the next vertex in Q.

Example: BFS Run



Example: Another BFS Run



The End


THE APOCALYPSE! THE SKIES BURN, THE SEAS TURN TO BLOOD, AND THE DEAD WALK THE EARTH!
THE DEAD WHAT?
WALK THE EARTH!
I HAVE TO GO.
SCRIBBLE SCRIBBLE
THE DEAD RETURN!
EVERYONE, QUICK, GET YOUR NAMES ON HERE!
MATH DEPT
AT LAST!
I HOPE THERE'S TIME!
HURRGHHH
CEMETARY
PAUL ERDÖS?
YES?
WE NEED YOU TO SIGN THIS.