Ma/CS 6a

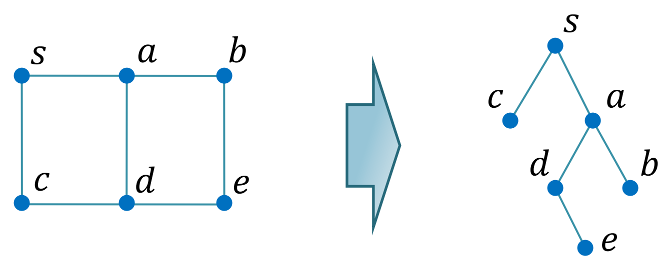
Class 7: More BFS

By Adam Sheffer

Six Degrees of Kevin Bacon

- **Problem.** Given a database of every actor and the movies that s\he played in, how can we compute everybody's Bacon numbers?
- We build the actors graph, and run the *BFS algorithm* from Kevin Bacon's vertex.

BFS Output

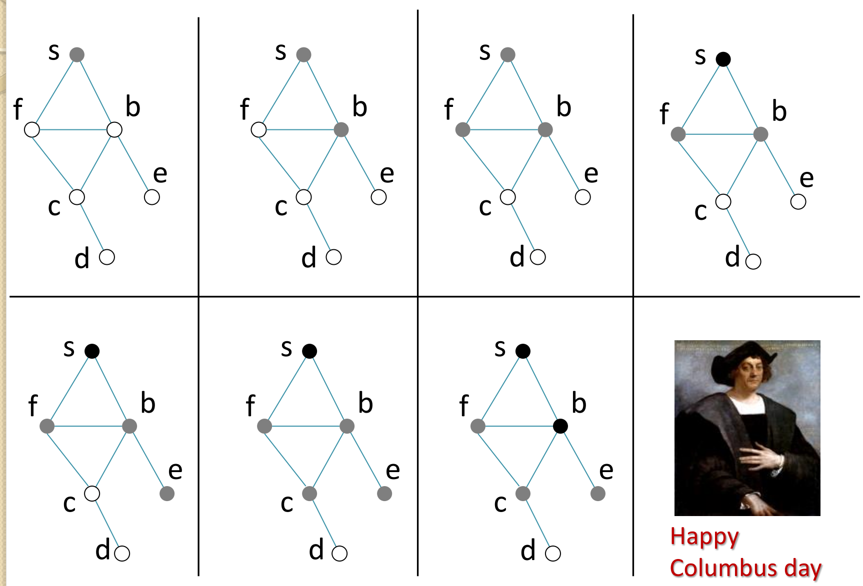


- The output is a *BFS tree*, containing only shortest paths from s.
  - A rooted tree with root s.

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



Happy
Columbus day

BFS: Pseudocode

Algorithm 1 BFS(G,s)

```
foreach  vertex u ∈ V − {s} do
    Color[u] ← White
    d[u] ← ∞
    π[u] ← NIL
Color[s] ← Gray
d[s] ← 0
π[s] ← NIL
Q ← {s}
while  Q ≠ ∅ do
    u ← Head[Q]
    foreach  v ∈ ADJ[u] do
        if  Color[v] = White then
            Color[v] ← Gray
            d[v] ← d[u] + 1
            π[v] ← u
            Insert v to Q
    Delete u from Q
    Color[u] ← Black
```

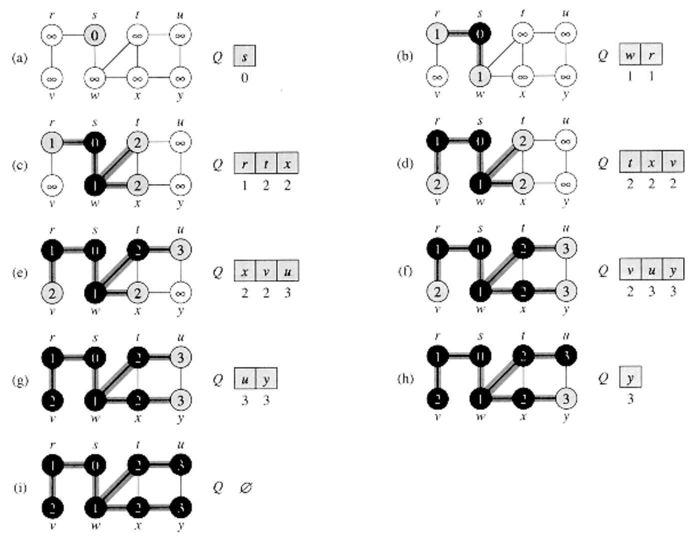
Placing the root in Q

Going over u's neighbors

Distance from the root

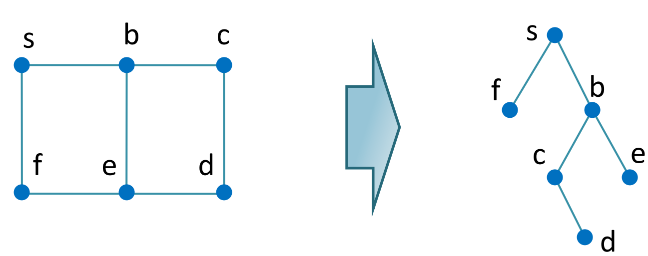
The parent of v

Example: Another BFS Run



The BFS Tree

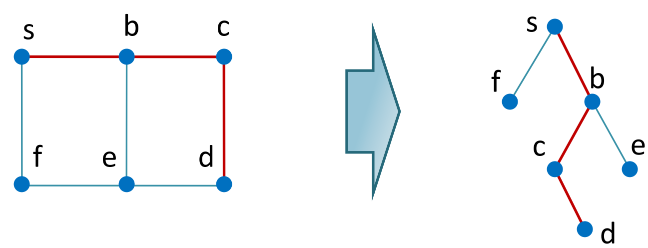
- We keep only edges that correspond to a π field of some vertex.
- We now prove that this results in a rooted tree of shortest paths from s.



Using the π values

- Using the π values, how can we find a shortest path between s and v?

$$v, \pi[v], \pi[\pi[v]], \cdots, s$$



Shortest Paths

- $\delta(s, v)$ – the length of the shortest path between s and v.
- If there's no path between these two vertices, then $\delta(s, v) = \infty$.
- To prove the correctness of the BFS algorithm, we need to prove that for every $v \in V$:

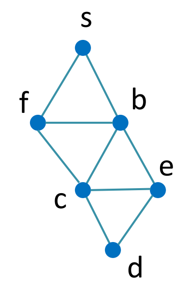
$$d[v] = \delta(s, v).$$

Warm-up Claim

- **Claim 1.** Given a graph $G = (V, E)$ and an edge $(u, v) \in E$, then
$$\delta(s, v) \leq \delta(s, u) + 1.$$
- **Examples.**

$2 = \delta(s, c) \leq \delta(s, e) + 1 = 3$

$3 = \delta(s, d) \leq \delta(s, c) + 1 = 3$

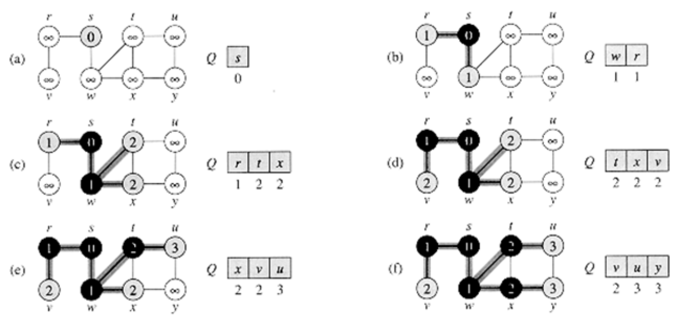


Proof of Warm-up Claim

- **Claim 1.** Given a graph $G = (V, E)$ and an edge $(u, v) \in E$, then
$$\delta(s, v) \leq \delta(s, u) + 1.$$
- **Proof.**
  - If there are no paths between s and u, then $\delta(s, v) = \delta(s, u) = \infty$.
  - Otherwise, there is a path of length $\delta(s, u)$ between s and u. Adding the edge (u, v) at the end of this path results in a path from s to v of length $\delta(s, u) + 1$.

$d[v] \geq \delta(s,v)$

- **Claim 2.** At every step of the BFS algorithm, and for every $v \in V$, we have $d[v] \geq \delta(s,v)$.



Proof of $d[v] \geq \delta(s,v)$

- **Proof.** By induction on the number of vertices discovered by the algorithm:
  - Basis: at first the claim holds since $d[s] = 0$ and $d[v] = \infty$ for any other $v \in V$.
  - Step: when discovering vertex v by examining an edge going out of u:
  - $d[v] = d[u] + 1 \underset{\text{Induction hypothesis}}{\geq} \delta(s,u) + 1 \underset{\text{Claim 1}}{\geq} \delta(s,v)$

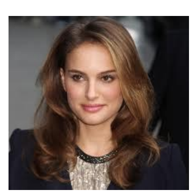
What is Left to Do?

- **Claim 2.** At every step of the BFS algorithm, and for every $v \in V$, we have
$$d[v] \geq \delta(s, v).$$
- How does this help us in proving the correctness of the BFS algorithm?
  - If suffices to claim that at the end of the algorithm $d[v] \leq \delta(s, v)$ for every v.

Erdős-Bacon Numbers

- The ***Erdős-Bacon number*** of a person is the sum of his/her Bacon number and Erdős number.

Stephen Hawking
E#=4 B#=3 E+B=7

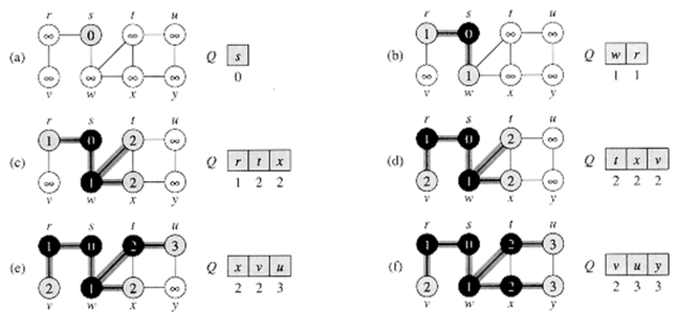


Natalie Portman
E#=6 B#=1 E+B=7

Daniel Kleitman
E#=1 B#=2 E+B=3

Vertices in the Queue

- **Claim 3.** At any step, if the queue Q consists of $v_1, v_2, \dots, v_i$ (in this order), then $d[v_1] \le d[v_2] \le \cdots \le d[v_i] \le d[v_1] + 1$.



Proof: Vertices in the Queue

- **Claim 3.** At any step, if the queue Q consists of $v_1, v_2, \dots, v_i$ (in this order), then $d[v_1] \le d[v_2] \le \cdots \le d[v_i] \le d[v_1] + 1$.
- **Proof.** By induction on the state of Q:
  - Basis: At first only s is in the queue.
  - Step: Assume that the claim holds up to now:
    - Removing the element at the front of Q is OK.
    - When inserting v at the back of Q, and having u at the front, by definition $d[v] = d[u] + 1$.

BFS Correctness

- **Theorem.** Running BFS on a graph G and vertex s sets the correct values of $d[v]$ and $\pi[v]$ for every vertex v.
- **Proof.**
  - If $\delta(s,v)=\infty$ then by claim 2 that $d[v] \geq \delta(s,v)=\infty$.
  - If we discover v during the algorithm, $\delta[v] \neq \infty$, so this never happens and $\pi[v]=\emptyset$.

BFS Correctness (cont.)

- The case where $\delta(s,v) \neq \infty$ is proved by induction on $\delta(s,v)$:
- Basis: The only vertex with $\delta=0$ is s.
- Step: Assume that this holds for $\delta(s,v) \leq k-1$ and consider a vertex u with $\delta(s,u)=k$.
  - By Claim 3 and $\delta(s,u)=k$, the alg. discovers u after discovering every v with $\delta(s,v)=k-1$.
  - Let v be the first vertex that was discovered such that there exists an edge (u,v) and $\delta(s,v)=k-1$.
  - The algorithm discovers v through u and sets $d[v]=d[u]+1=k$ and $\pi[v]=u$.

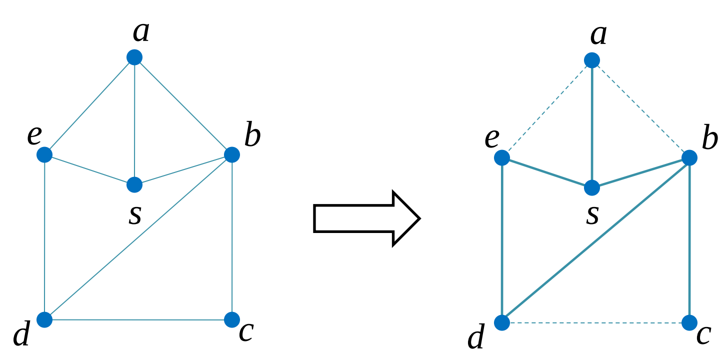
Problem: Shortest Paths Graph

- Given a graph $G = (V, E)$ and a vertex $s \in V$, the ***shortest paths graph*** of s is the graph $G' = (V, E')$, where

$$E' = \{e \in E \mid e \text{ is part of a shortest path from } s\}.$$

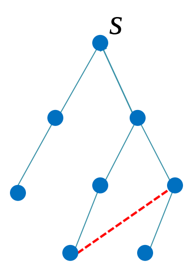
- **Problem.** Given an undirected graph $\mathrm{G} = (V, E)$ and a vertex $s \in V$, find an efficient algorithm for computing the shortest paths graph of s.

Example: Shortest Paths Graph



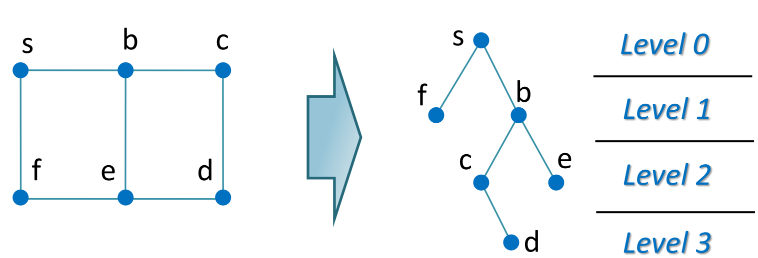
Solution

- Run the BFS algorithm from s, to obtain a shortest paths tree from s.
- There might be shortest paths that are not in this tree!



Levels of the BFS Tree

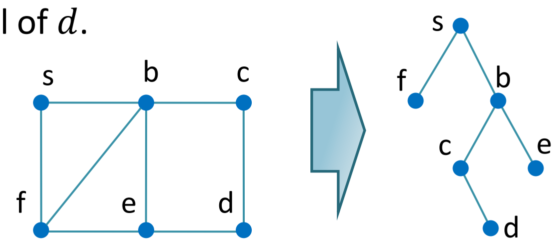
- The ***i'th level*** of the BFS tree is the set of vertices $v \in V$ that satisfy $d(v) = i$.



* This is the origin of the name ***B***readth ***F***irst ***S***earch.

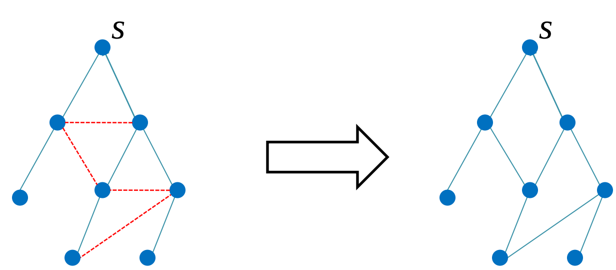
Graph Edges not in the BFS Tree

- A graph edge that is not in the BFS tree:
  - *Can be* between vertices of the same level.
  - *Can be* between vertices in consecutive levels.
  - *Cannot be* between vertices not in consecutive levels. That would mean that the lower vertex is in the wrong level.
  - **Example.** Adding the edge (b, d) changes the level of d.



Solution – Second Attempt

- Change the BFS algorithm so that it only removes edges between vertices of the same level.

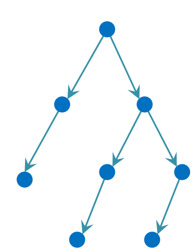


Correctness of the Solution

- **Prove.** An edge e between vertices u, v of respective levels $i-1, i$ is on a shortest path:
  - The length of any shortest path to v is i.
  - There is a path P of length $i-1$ between s and u.
  - Connecting e to the end of P yields a path of length i to v (i.e., a shortest path).
- We can similarly show that an edge between vertices of the same level cannot participate in a shortest path.

BFS in a Directed Graph

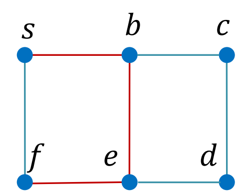
- In a *directed graph*, a path has to be in the direction of the edges.
- It's not hard to verify that BFS also works for directed graphs.
  - When taking a vertex v out of the queue, consider only edges that come out of v.
- Another difference is that the edges of the BFS tree are directed down.
  - More in Assignment 3…



Another Question, If Time Permits

- **Problem.** Consider a graph $G = (V, E)$ and a vertex $s \in V$. Additionally, every edge is colored either red or blue.
  - We redefine the length of a path as the number of blue edges in it.
  - Perform a *small* change in the BFS algorithm, to work according to this new definition.

$$\delta(s, f) = 0$$
$$\delta(s, d) = 1$$



The End

- *Sabbath number* - a series of musical collaborations to get to the rock band Black Sabbath.
- *Erdős-Bacon-Sabbath Numbers:*

	Erdős	Bacon	Sabbath	E-B-S
Richard Feynman	3	3	4	10
Carl Sagan	4	2	3	9
Terry Pratchett	4	2	3	9
Condoleeza Rice	6	3	4	13
Buzz Aldrin	6	2	3	11
Natalie Portman	5	2	4	11