

Lecture 14

CSE 331

Feb 26, 2020

Breadth First Search (BFS)

Build layers of vertices connected to s

$$L_0 = \{s\}$$

Assume L_0, \dots, 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

Use linked lists

Use $CC[v]$ array

Rest of Today's agenda

Quick run time analysis for BFS

Quick run time analysis for DFS (and Queue version of BFS)

Helping you schedule your activities for the day

$O(m+n)$ BFS Implementation

BFS(s)

Array

Input graph as
Adjacency list

$CC[s] = T$ and $CC[w] = F$ for every $w \neq s$

Set $i = 0$

Set $L_0 = \{s\}$

While L_i is not empty

$L_{i+1} = \emptyset$

For every u in L_i

For every edge (u, w)

If $CC[w] = F$ then

$CC[w] = T$

Add w to L_{i+1}

$i++$

Linked List

Version in KT
also
computes a
BFS tree

All the layers as one

BFS(*s*)

$CC[s] = T$ and $CC[w] = F$ for every $w \neq s$

Set $i = 0$

Set $L_0 = \{s\}$

While L_i is not empty

$L_{i+1} = \emptyset$

For every u in L_i

For every edge (u, w)

If $CC[w] = F$ then

$CC[w] = T$

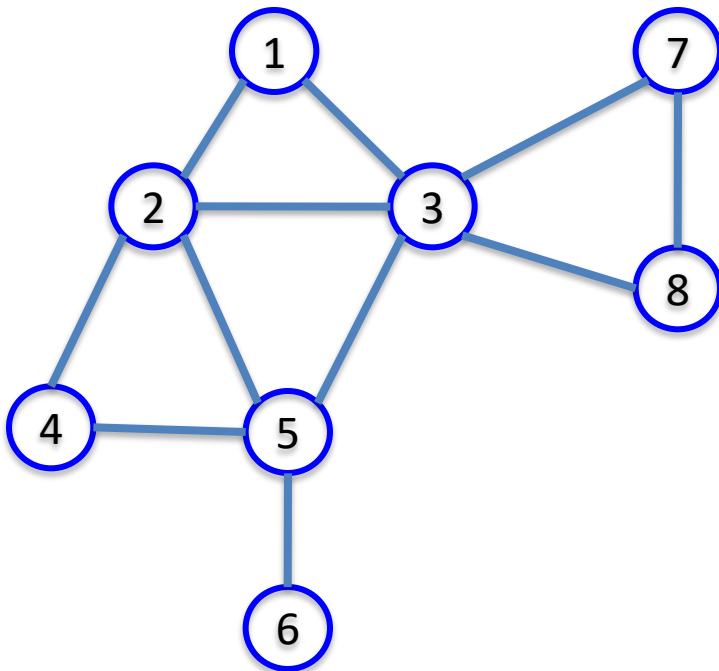
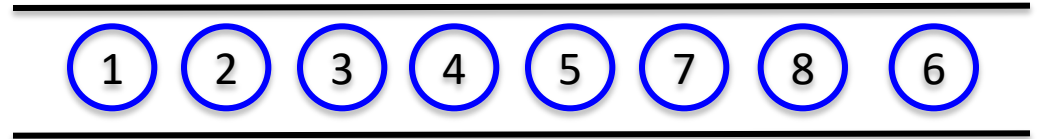
Add w to L_{i+1}

$i++$

All layers are considered in first-in-first-out order

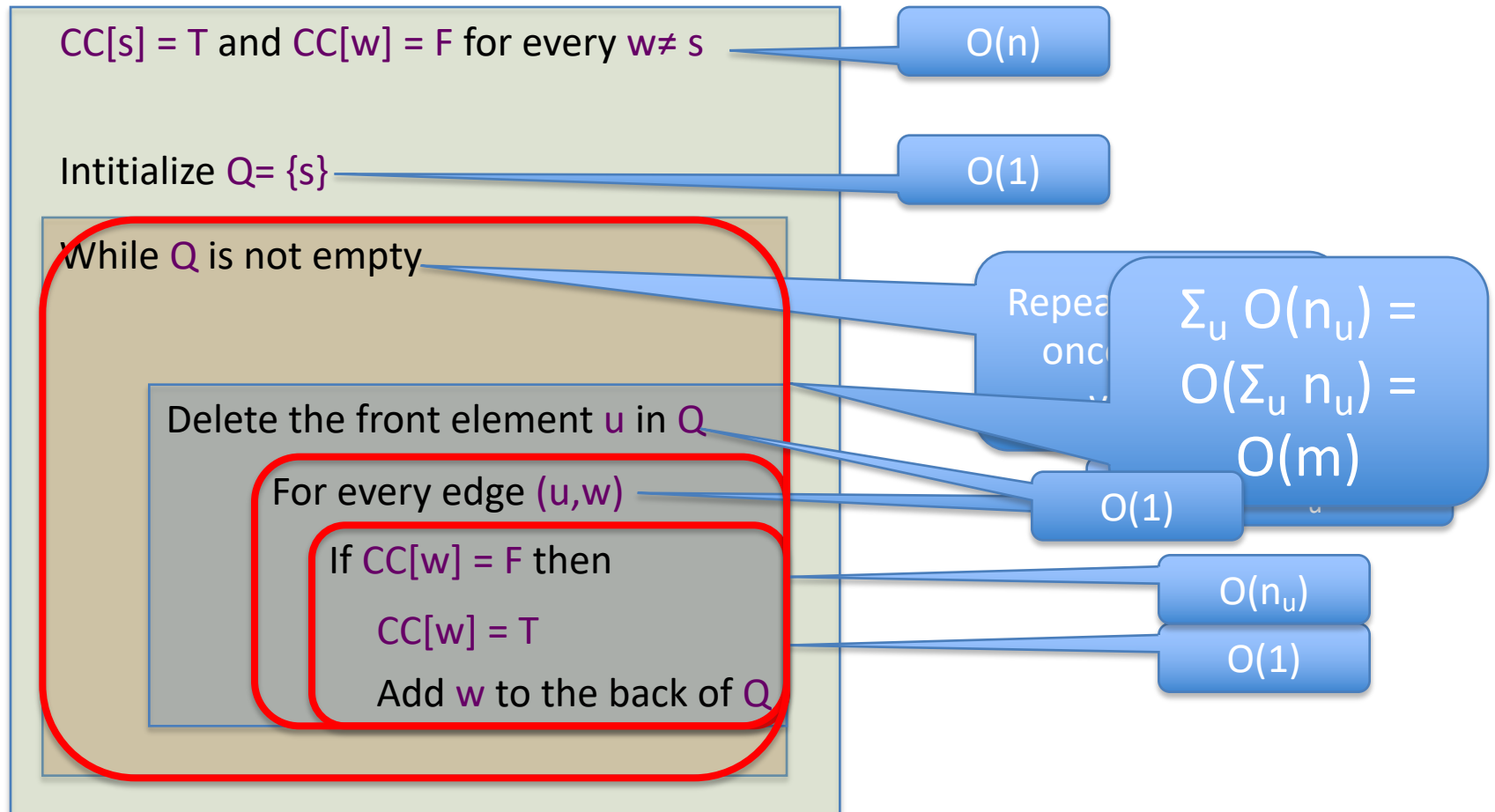
Can combine all layers into one queue: all the children of a node are added to the end of the queue

An illustration



Queue $O(m+n)$ implementation

BFS(s)

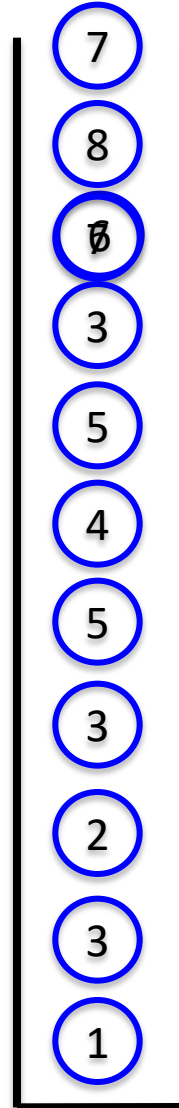
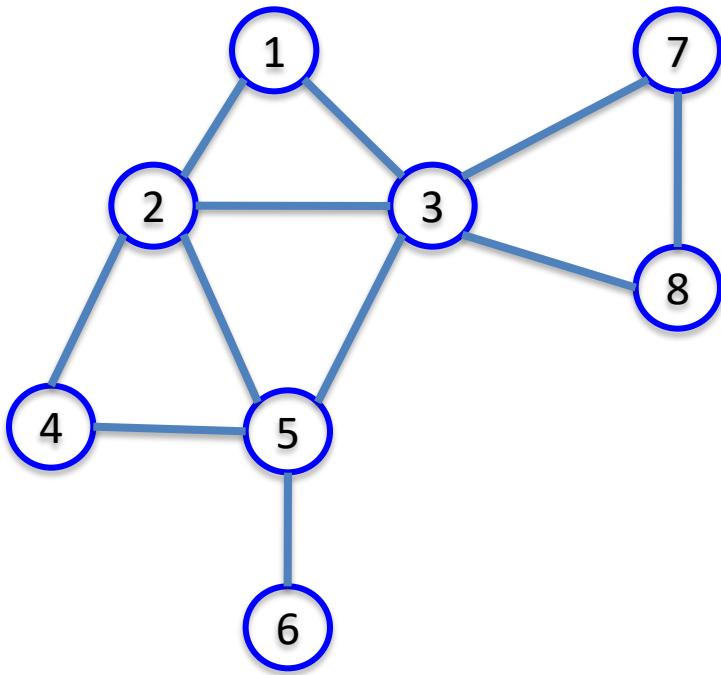


Questions?

Implementing DFS in $O(m+n)$ time

Same as BFS except stack instead of a queue

A DFS run using an explicit stack



DFS stack implementation

DFS(s)

$CC[s] = T$ and $CC[w] = F$ for every $w \neq s$

Initialize $\hat{S} = \{s\}$

While \hat{S} is not empty

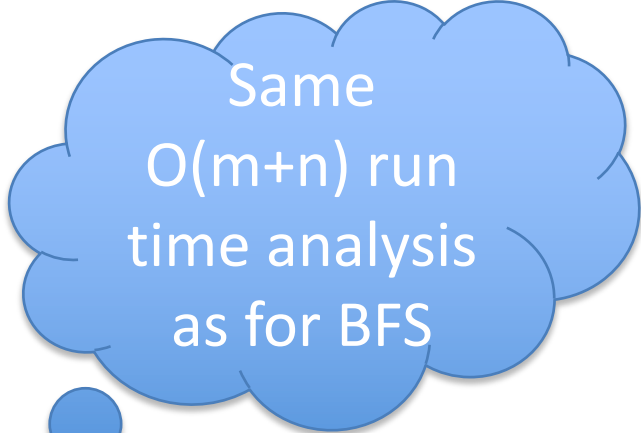
Pop the top element u in \hat{S}

For every edge (u, w)

If $CC[w] = F$ then

$CC[w] = T$

Push w to the top of \hat{S}



Same
 $O(m+n)$ run
time analysis
as for BFS

Questions?

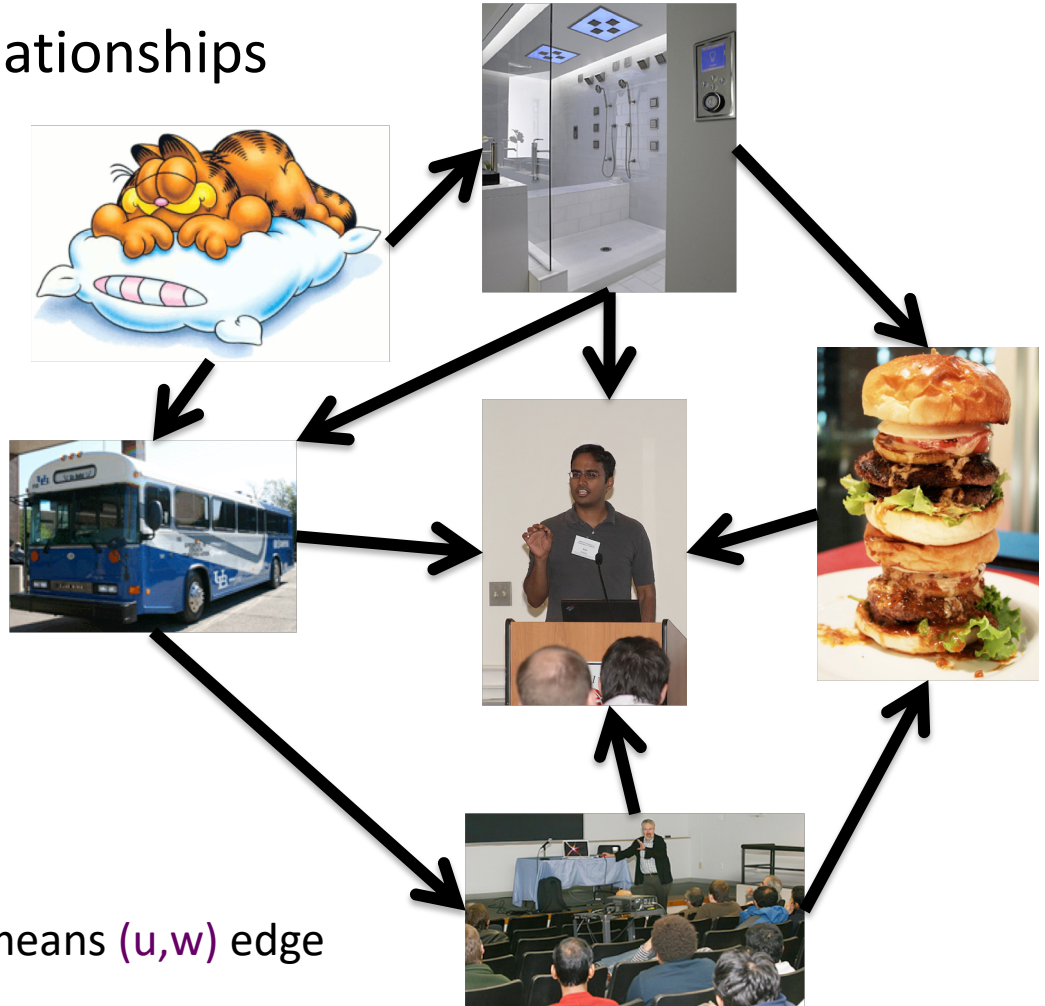
Reading Assignment

Sec 3.3, 3.4, 3.5 and 3.6 of [KT]

Directed graphs

Model asymmetric relationships

Precedence relationships

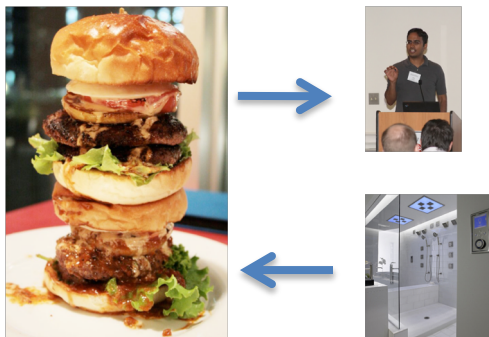
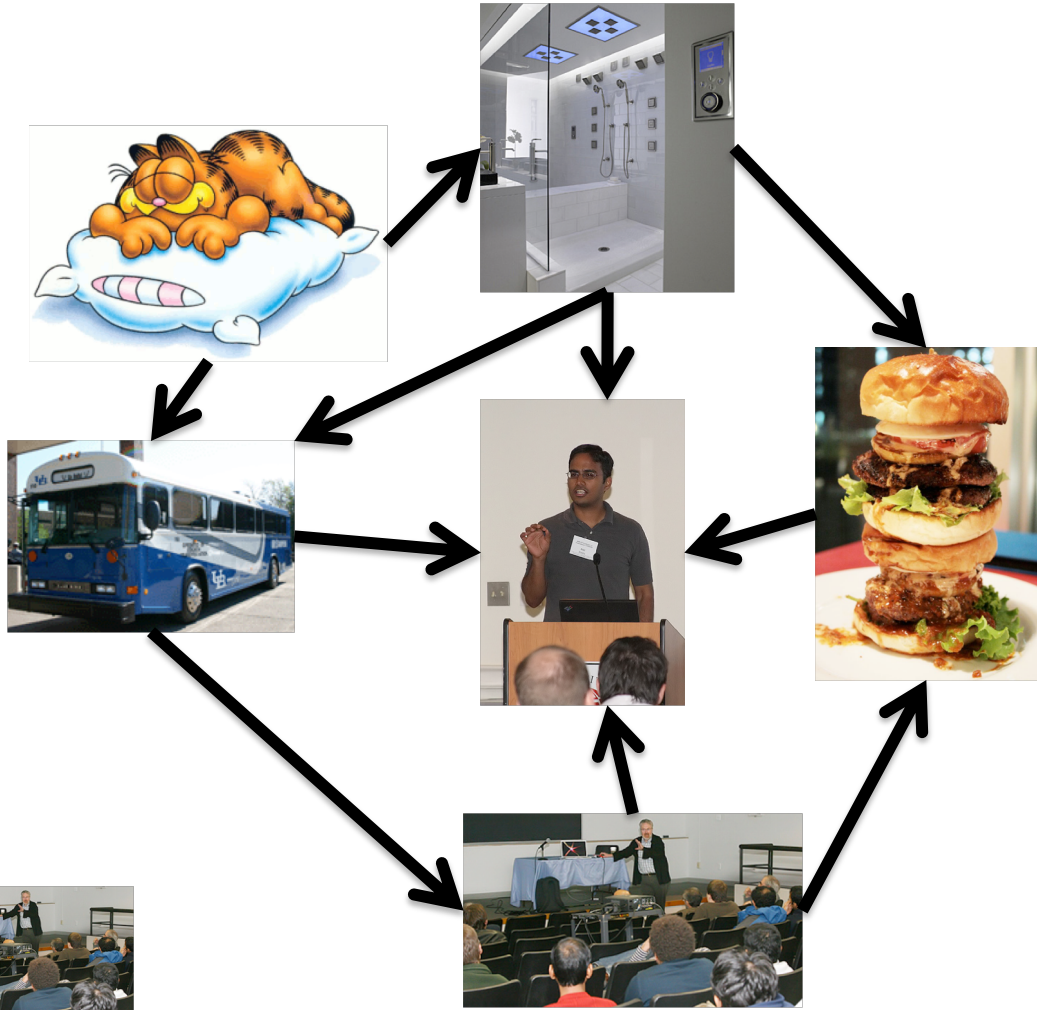


u needs to be done before w means (u,w) edge

Directed graphs

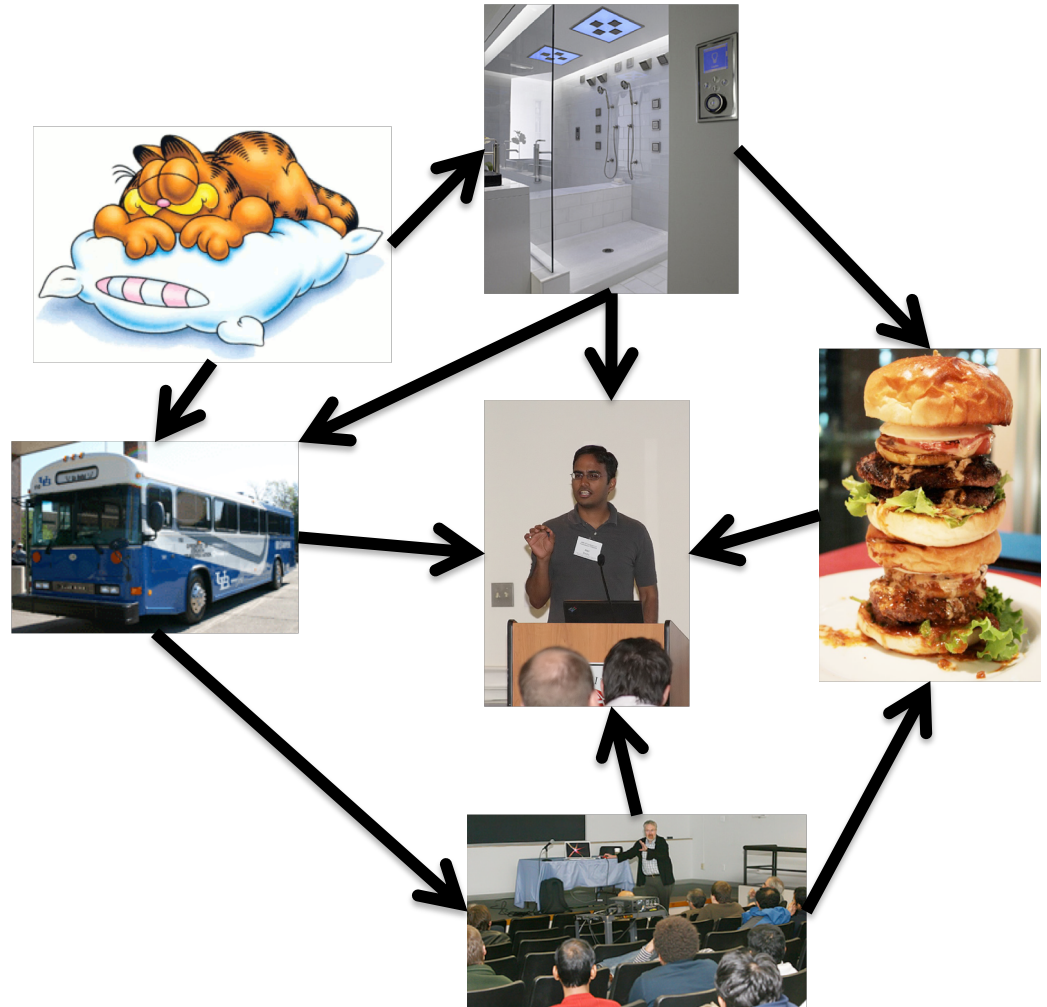
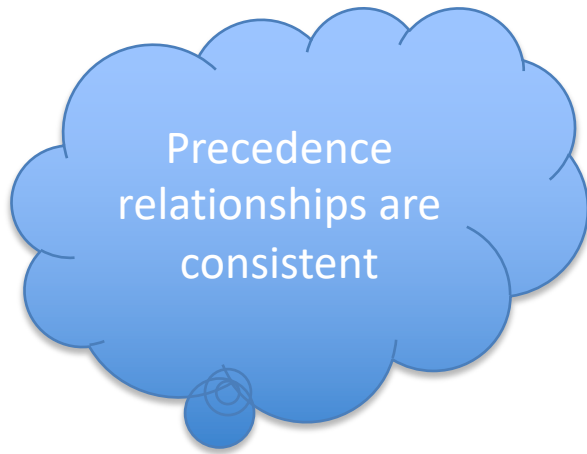
Adjacency matrix is not symmetric

Each vertex has two lists in Adj. list rep.



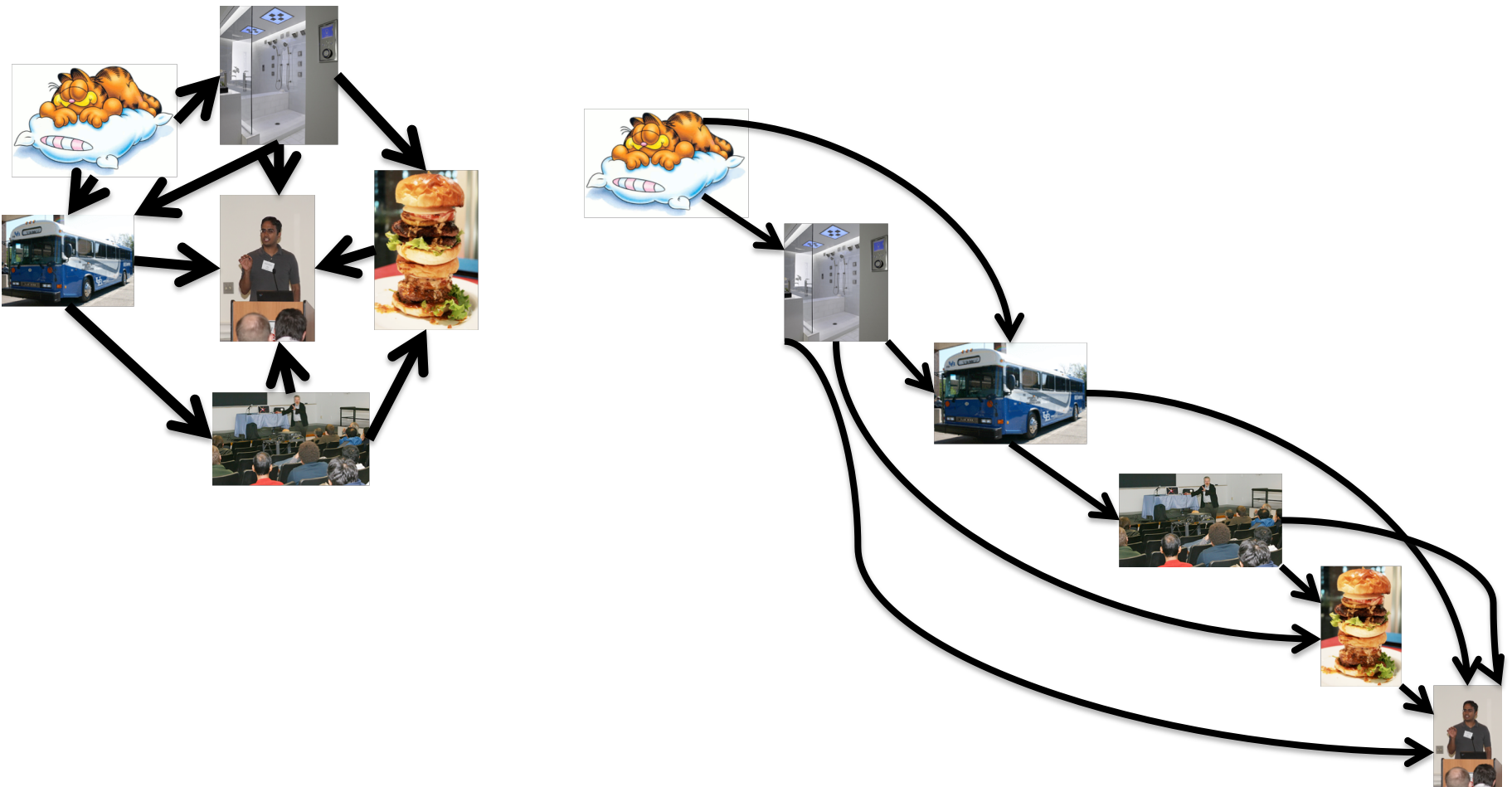
Directed Acyclic Graph (DAG)

No directed cycles



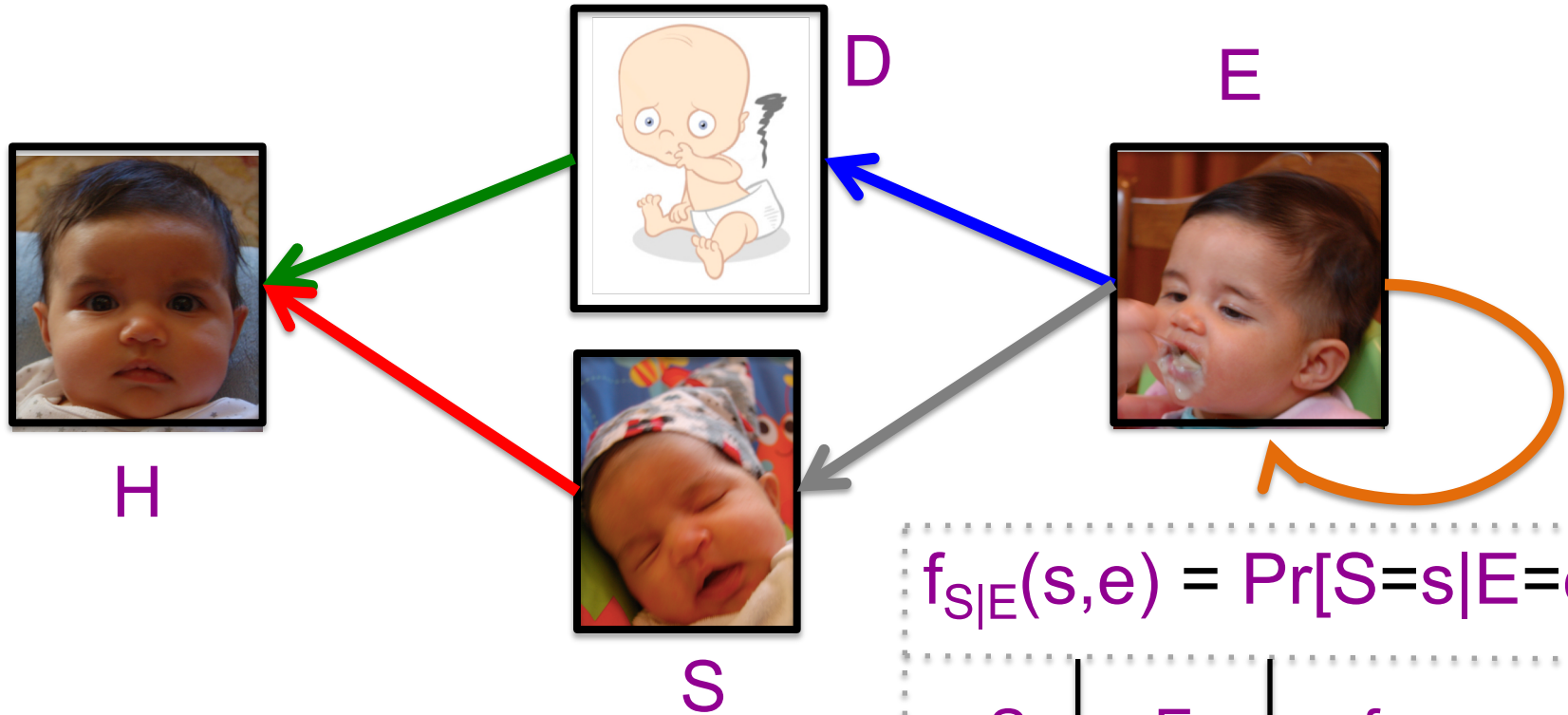
Topological Sorting of a DAG

Order the vertices so that all edges go “forward”



Probabilistic Graphical Models (PGMs)

<http://ginaskokopelli.com/wp-content/uploads/2013/01/DiaperDealsLogo.jpg>



$$f_{S|E}(s,e) = \Pr[S=s|E=e]$$

S	E	$f_{S E}$
1	1	0.8
1	0	0.3
0	1	0.2
0	0	0.7

$$\varphi(h) = \sum_{d,s,e} f_{H|D,S}(h,d,s) \times f_{S|E}(s,e) \times f_{D|E}(d,e) \times f_E(e)$$

More details on Topological sort

Topological Ordering

This page collects material from previous incarnations of CSE 331 on topological ordering.

Where does the textbook talk about this?

[Section 3.6](#) in the textbook has the lowdown on topological ordering.

Fall 2018 material

First lecture

Here is the lecture video:

CSE331 on 10/1/2018 (Mon)



Questions?