

Today's announcements:

MP7 available. Due 12/6, 11:59p. EC due 11/30, 11:59p.

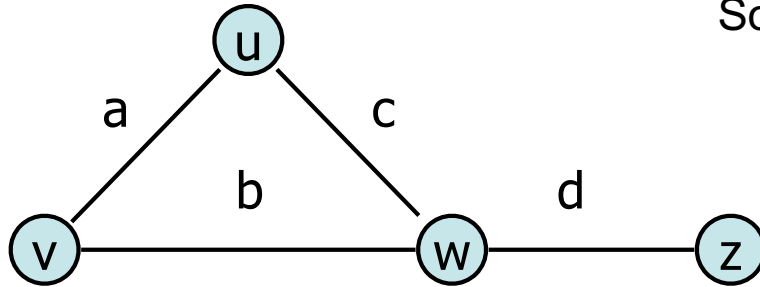
Exam5: 12/4-12/7, CBTF.

How do we get from here to there?

Need:

1. Common Vocabulary
2. Graph implementation
3. Traversal
4. Algorithms.

Graphs: Adjacency Matrix



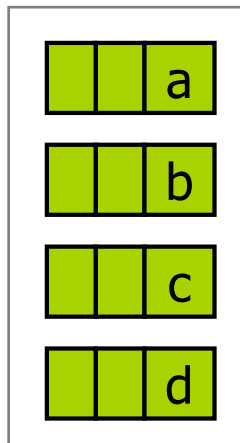
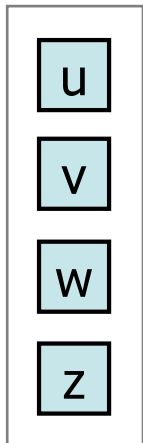
Some functions we'll compare:

`insertVertex(vertex v)`

`removeVertex(vertex v)`

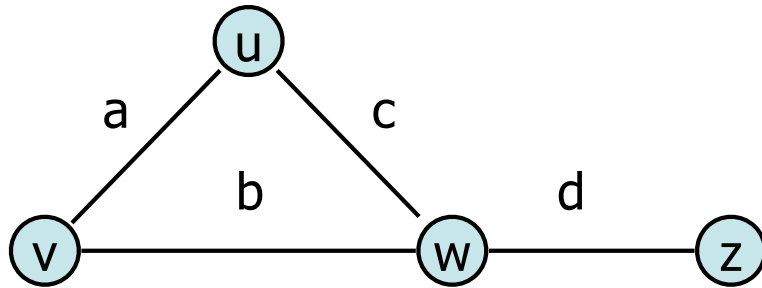
`areAdjacent(vertex v, vertex u)`

`incidentEdges(vertex v)`



	u	v	w	z
u				
v				
w				
z				

Graphs: Adjacency List



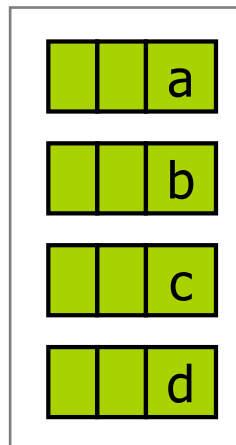
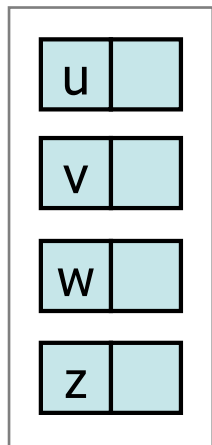
Some functions we'll compare:

`insertVertex(vertex v)`

`removeVertex(vertex v)`

`areAdjacent(vertex v, vertex u)`

`incidentEdges(vertex v)`



Graphs: Asymptotic Performance

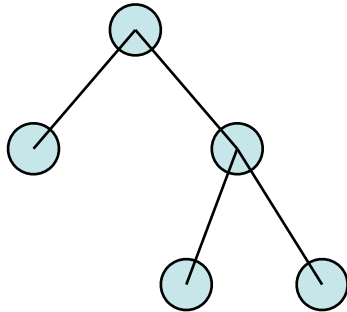
• n vertices, m edges • no parallel edges • no self-loops • Bounds are big-O	Edge List	Adjacency List	Adjacency Matrix
Space	$n + m$	$n + m$	n^2
incidentEdges(v)	m	deg(v)	n
areAdjacent (v , w)	m	min(deg(v), deg(w))	1
insertVertex(o)	1	1	n^2
insertEdge(v , w , o)	1	1	1
removeVertex(v)	m	deg(v)	n^2
removeEdge(e)	1	1	1

Graphs – traversal

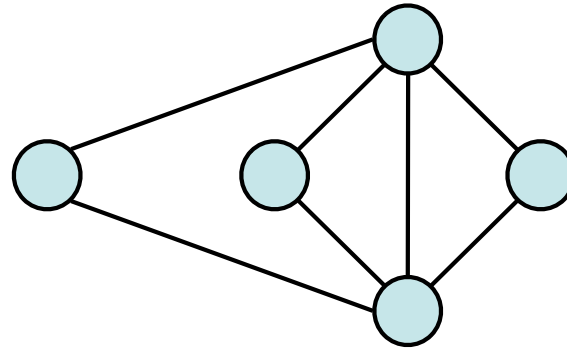
Objective: Visit every vertex and every edge, in the graph,

Purpose: We can search for interesting substructures in the graph,

Contrast graph traversal to BST traversal:



- Ordered
- Obvious start

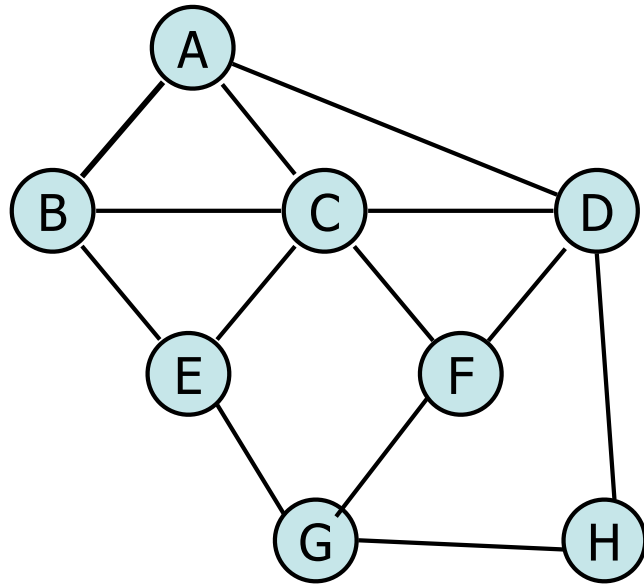


•

•

•

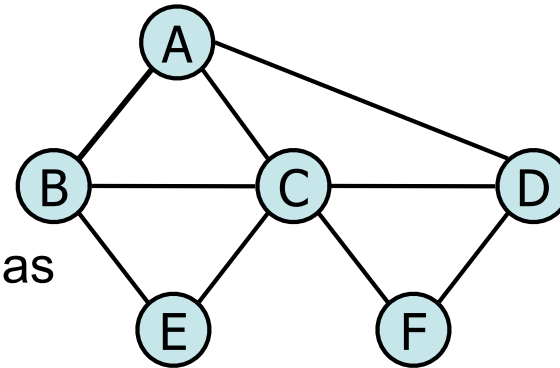
Graphs: BFS example



A	C B D
B	A C E
C	B A D E F
D	A C F H
E	B C G
F	C D G
G	E F H
H	D G

Graphs: Traversal – BFS

Visits every vertex and classifies each edge as either “discovery” or “cross”



Algorithm BFS(G)

Input: graph G

Output: labeling of the edges of G as discovery edges and back edges

For all u in G.vertices()

 setLabel(u, UNEXPLORED)

For all e in G.edges()

 setLabel(e, UNEXPLORED)

For all v in G.vertices()

 if getLabel(v) = UNEXPLORED

 BFS(G,v)

Algorithm BFS(G,v)

Input: graph G and start vertex v

Output: labeling of the edges of G in the connected component of v as discovery edges and cross edges

queue q;

setLabel(v, VISITED)

q.enqueue(v);

While !(q.isEmpty)

 q.dequeue(v)

 For all w in G.adjacentVertices(v)

 if getLabel(w) = UNEXPLORED

 setLabel((v,w),DISCOVERY)

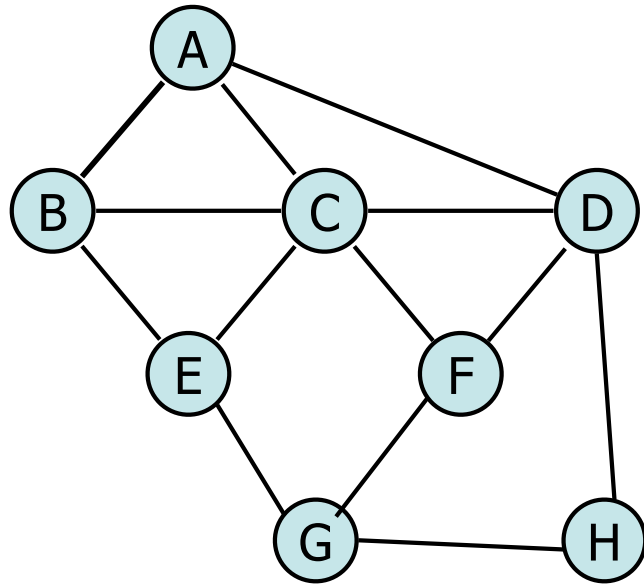
 setLabel(w, VISITED)

 q.enqueue(w)

 else if getLabel((v,w)) = UNEXPLORED

 setLabel((v,w),CROSS)

Graphs: BFS example



A	C B D
B	A C E
C	B A D E F
D	A C F H
E	B C G
F	C D G
G	E F H
H	D G

While loop

For loop

TOTAL RUNNING TIME: