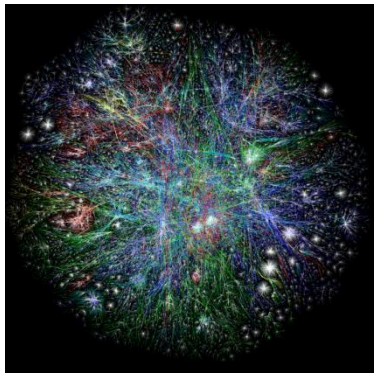
CS 225

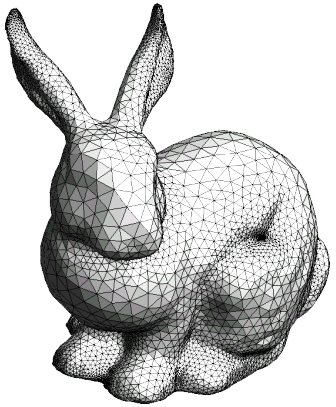
Data Structures

April 13 – Graph Impl

Wade Fagen-Ulmschneider

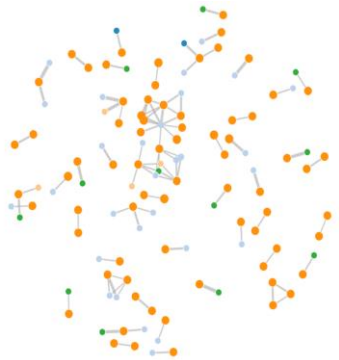
Graphs

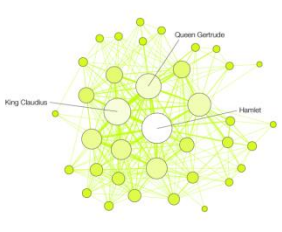




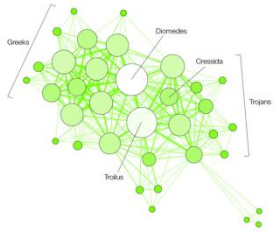
To study all of these structures:

1. A common vocabulary
2. Graph implementations
3. Graph traversals
4. Graph algorithms

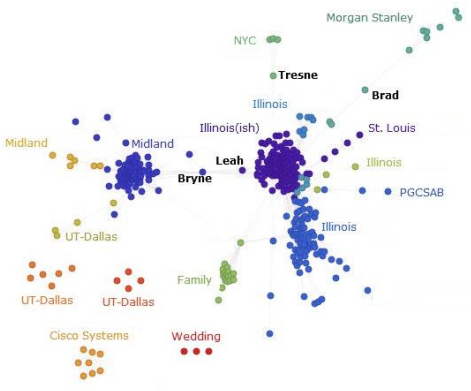


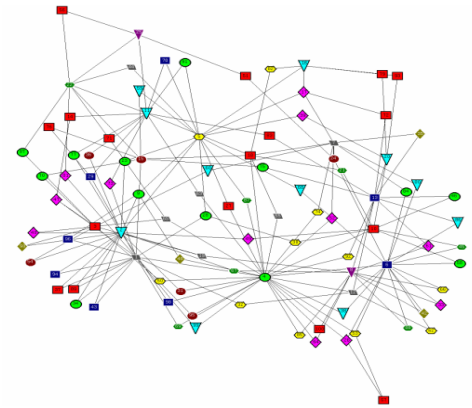


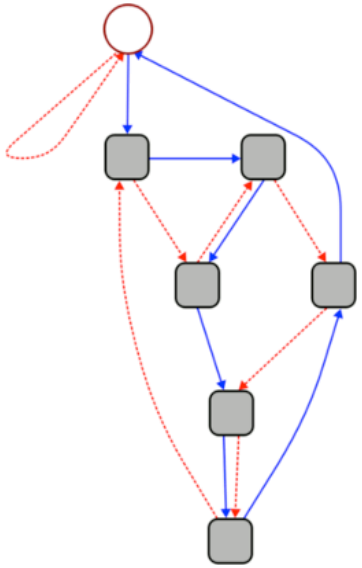
HAMLET

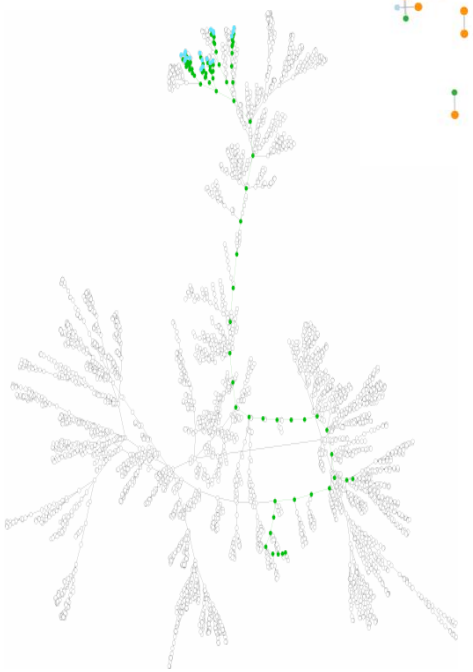


TROILUS AND CRESSIDA





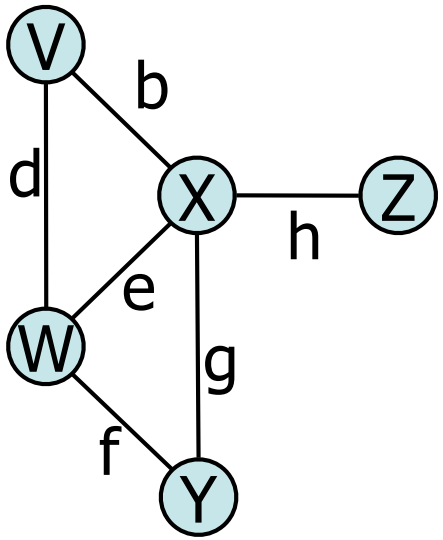




Graph ADT

Data:

- Vertices
- Edges
- Some data structure maintaining the structure between vertices and edges.



Functions:

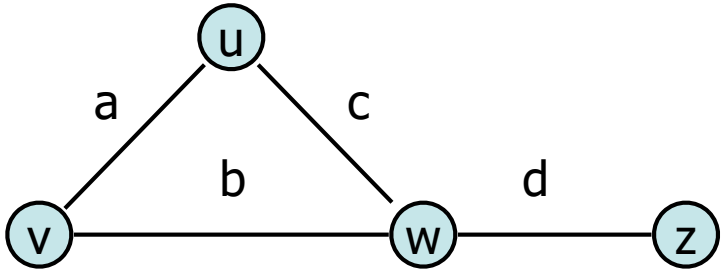
- insertVertex(K key);
- insertEdge(Vertex v1, Vertex v2, K key);

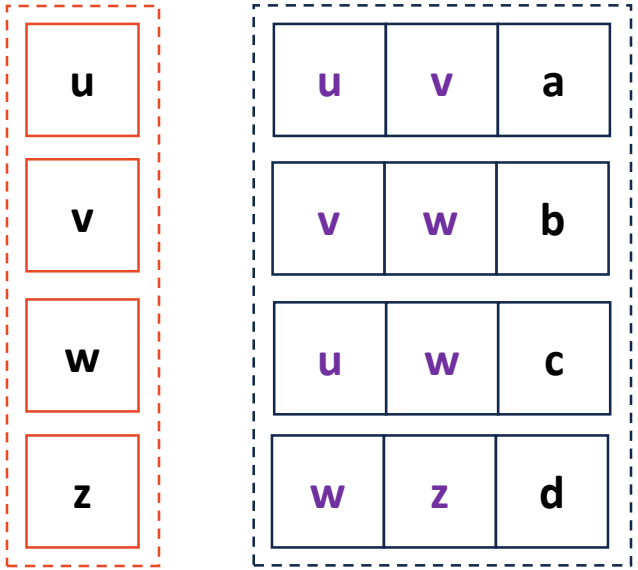
- removeVertex(Vertex v);
- removeEdge(Vertex v1, Vertex v2);

- incidentEdges(Vertex v);
- areAdjacent(Vertex v1, Vertex v2);

- origin(Edge e);
- destination(Edge e);

Graph Implementation: Edge List





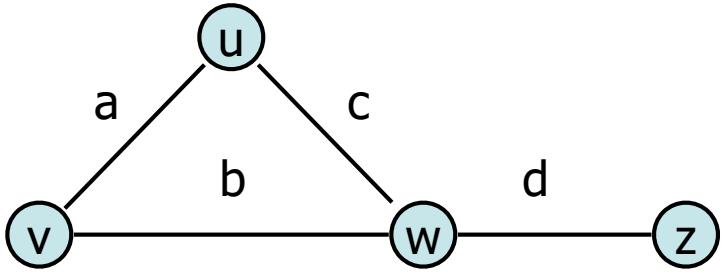
insertVertex(K key);

removeVertex(Vertex v);

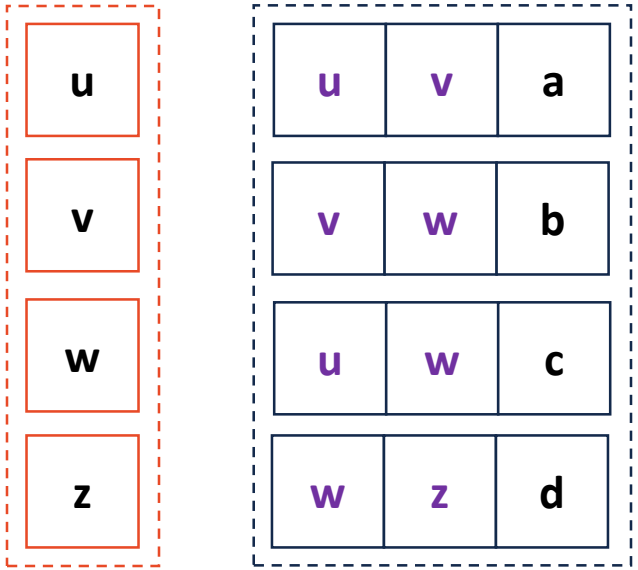
areAdjacent(Vertex v1, Vertex v2);

incidentEdges(Vertex v);

Graph Implementation: Adjacency Matrix

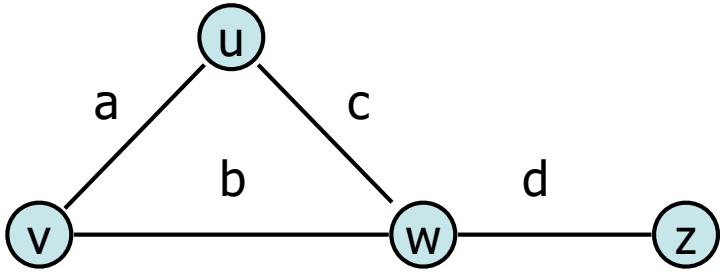


insertVertex(K key) :
removeVertex(Vertex v) :
areAdjacent(Vertex v1, Vertex v2) :
incidentEdges(Vertex v) :

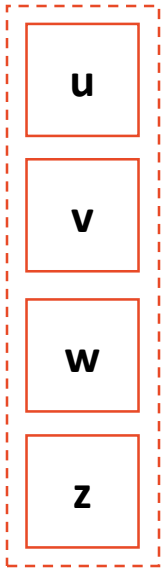


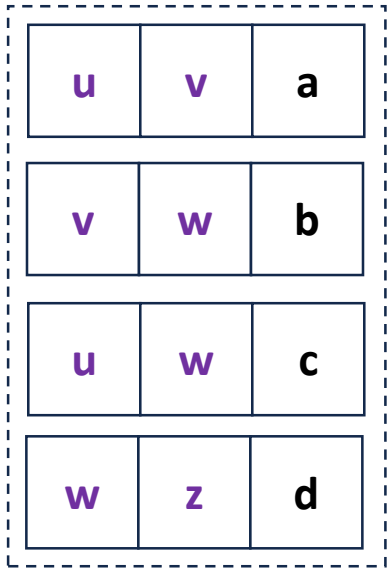
	u	v	w	z
u				
v				
w				
z				

Graph Implementation: Edge List



insertVertex(K key);
removeVertex(Vertex v);
areAdjacent(Vertex v1, Vertex v2);
incidentEdges(Vertex v);





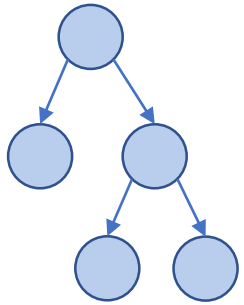
Expressed as O(f)	Edge List	Adjacency Matrix	Adjacency List
Space	n+m	n+m	n^2
insertVertex(v)	1	n	1
removeVertex(v)	m	n	deg(v)
insertEdge(v, w, k)	1	1	1
removeEdge(v, w)	1	1	1
incidentEdges(v)	m	n	deg(v)
areAdjacent(v, w)	m	1	min(deg(v), deg(w))

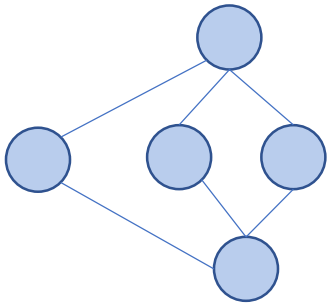
Traversal:

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

Purpose: Search for interesting sub-structures in the graph.

We've seen traversal before ….but it's different:





- Ordered
- Obvious Start
-

-
-
-

Traversal: BFS