



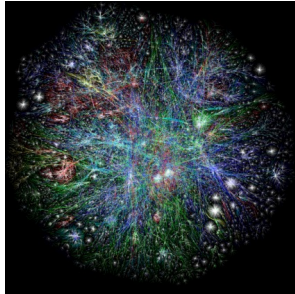
# CS 225

## Data Structures

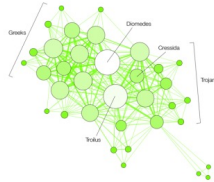
*March 20 – Graph Implementations*

*G Carl Evans*

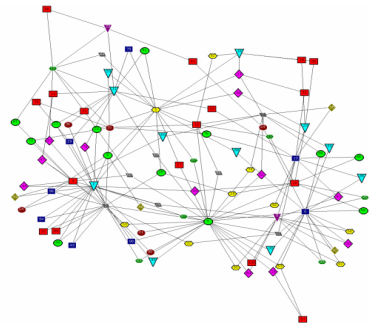
# Graphs



HAMLET

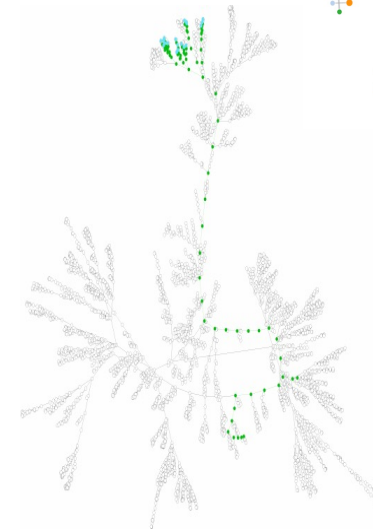
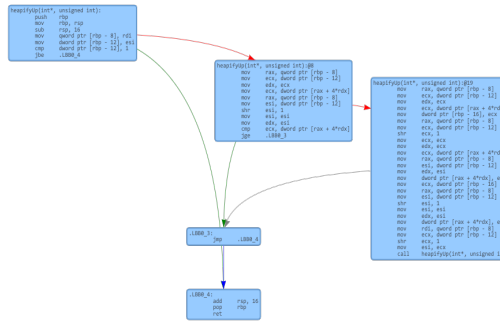
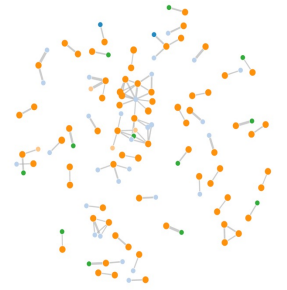
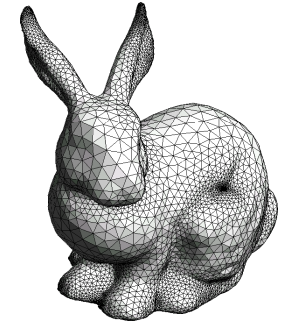


TROIUS AND CRESSIDA



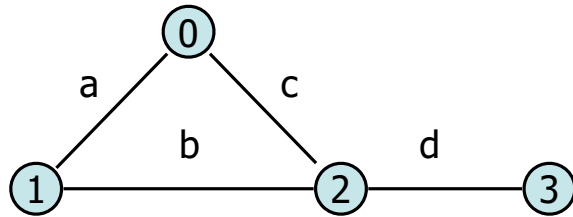
**To study all of these structures:**

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



# Graph Implementation: Edge List

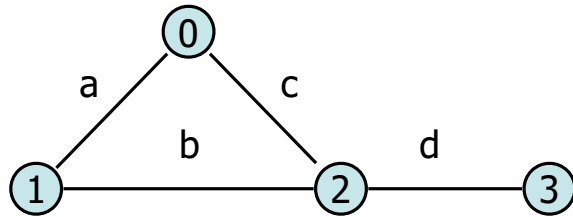
Vertex Collection:



0	0	1	a
1	1	2	b
2	0	2	c
3	2	3	d

Edge Collection:

# Graph Implementation: Edge List

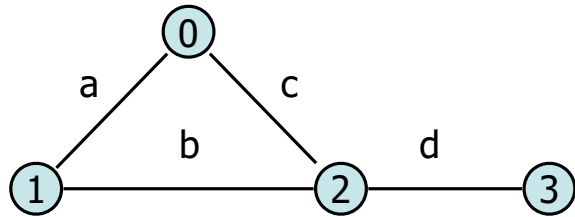


**insertVertex(K key):**

**removeVertex(Vertex v):**

0	0	1	a
1	1	2	b
2	0	2	c
3	2	3	d

# Graph Implementation: Edge List



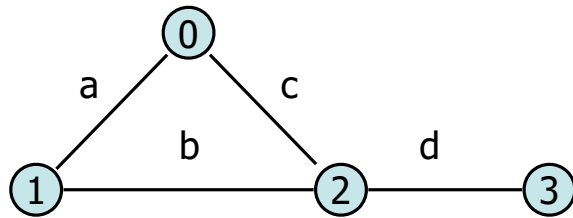
0	0	1	a
1	1	2	b
2	0	2	c
3	2	3	d

**incidentEdges(Vertex v):**

**areAdjacent(Vertex v1, Vertex v2):**

`G.incidentEdges(v1).contains(v2)`

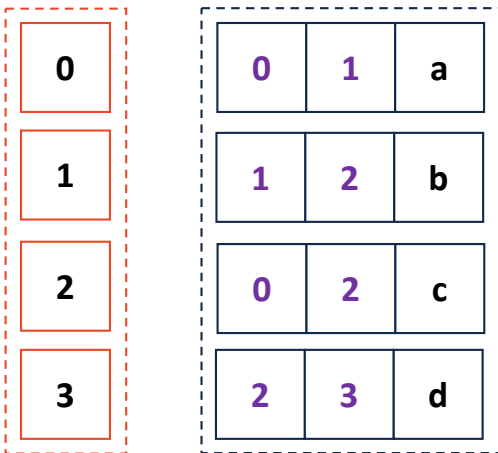
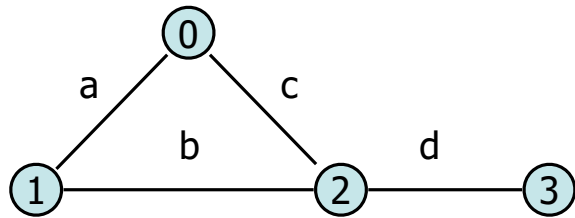
# Graph Implementation: Edge List



**insertEdge(Vertex v1, Vertex v2, K key):**

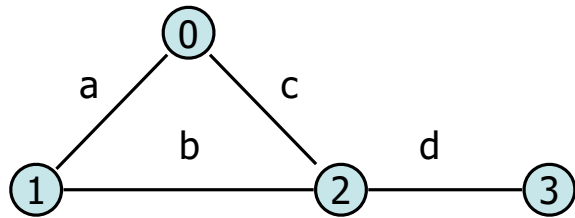
0	0	1	a
1	1	2	b
2	0	2	c
3	2	3	d

# Graph Implementation: Adjacency Matrix



	0	1	2	3
0				
1				
2				
3				

# Graph Implementation: Adjacency Matrix

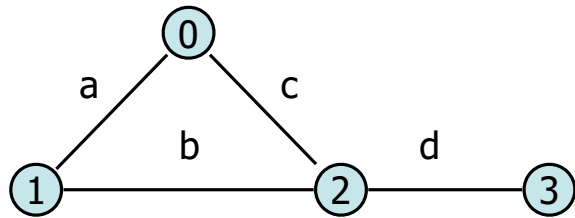


0	0	1	a
1	1	2	b
2	0	2	c
3	2	3	d

	0	1	2	3
0				
1				
2				
3				



# Graph Implementation: Adjacency Matrix

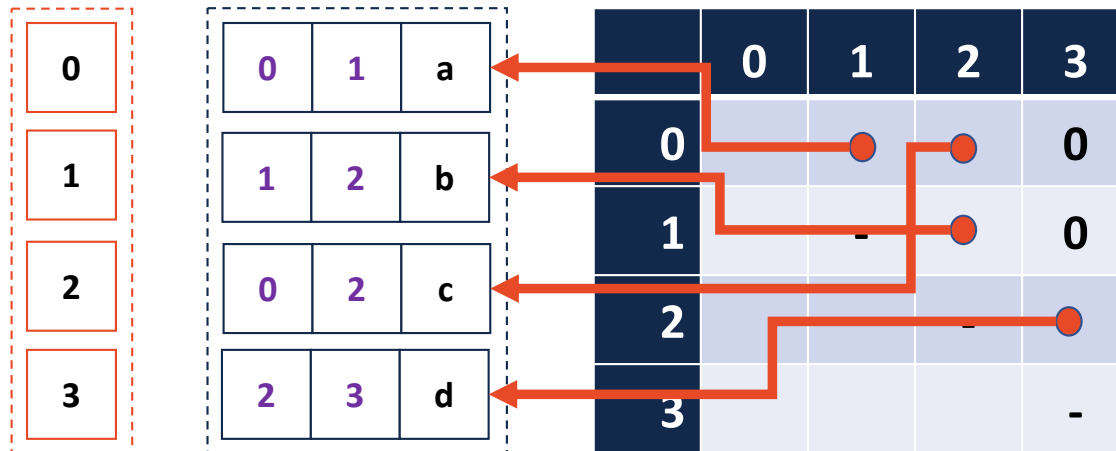
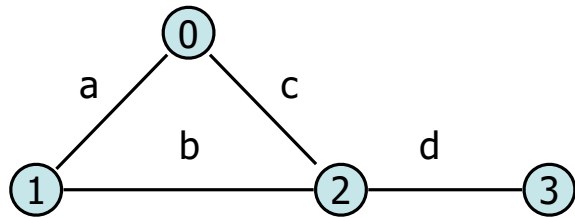


0
1
2
3

0	1	a
1	2	b
0	2	c
2	3	d

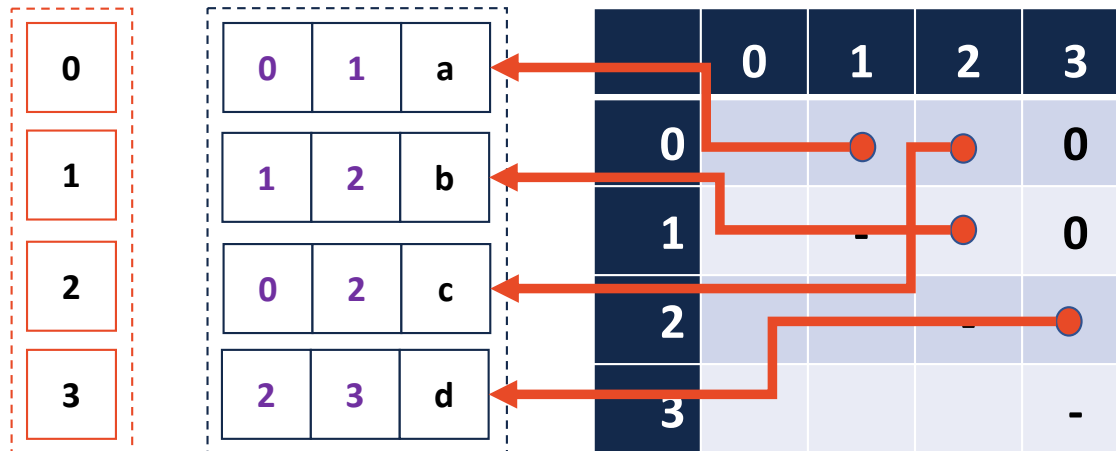
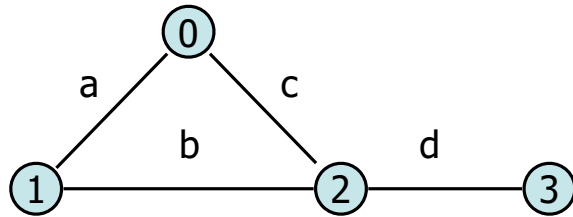
	0	1	2	3
0	-	1	1	0
1		-	1	0
2			-	1
3				-

# Graph Implementation: Adjacency Matrix



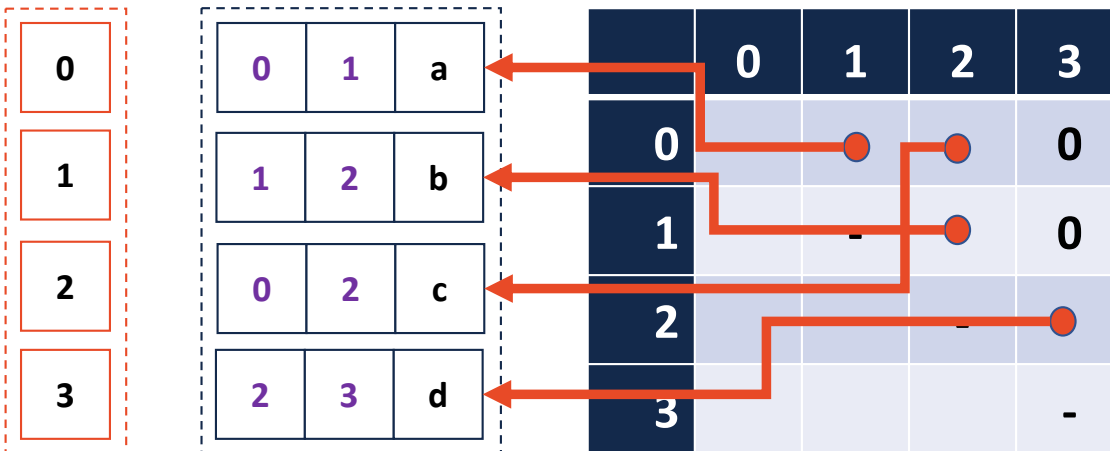
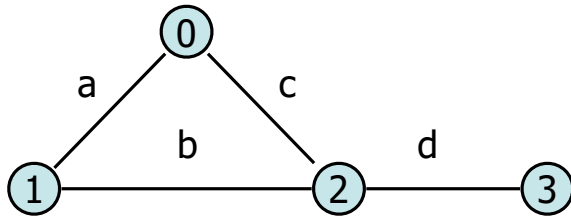
# Graph Implementation: Adjacency Matrix

**incidentEdges(Vertex v):**



# Graph Implementation: Adjacency Matrix

**areAdjacent(Vertex v1, Vertex v2):**



# Graph Implementation: Adjacency Matrix

**insertEdge(Vertex v1, Vertex v2, K key):**

