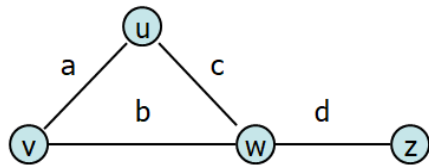


### Graph Implementation #1: Edge List

Vert.	Edges
u	
v	
w	
z	



#### Data Structures:

Vertex Collection:

Edge Collection:

#### Operations on an Edge List implementation:

insertVertex(K key):

- What needs to be done?

removeVertex(Vertex v):

- What needs to be done?

incidentEdges(Vertex v):

- What needs to be done?

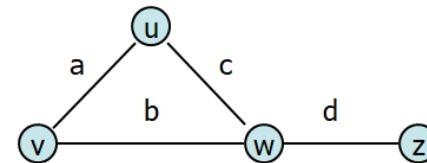
areAdjacent(Vertex v1, Vertex v2):

- Can this be faster than `G.incidentEdges(v1).contains(v2)`?

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

- What needs to be done?

### Graph Implementation #2: Adjacency Matrix



Vert.	Edges	Adj. Matrix
u		
v		
w		
z		

#### Data Structures:

#### Operations on an Adjacency Matrix implementation:

insertVertex(K key):

- What needs to be done?

removeVertex(Vertex v):

- What needs to be done?

incidentEdges(Vertex v):

- What needs to be done?

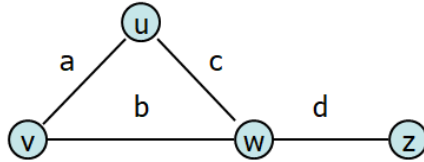
areAdjacent(Vertex v1, Vertex v2):

- Can this be faster than `G.incidentEdges(v1).contains(v2)`?

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

- What needs to be done?

### Graph Implementation #3: Adjacency List



Vertex List	Edges
u	
v	a
w	b, c
z	d

#### Operations on an Adjacency Matrix implementation:

insertVertex(K key):

removeVertex(Vertex v):

incidentEdges(Vertex v):

areAdjacent(Vertex v1, Vertex v2):

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

### Running Times of Classical Graph Implementations

	Edge List	Adj. Matrix	Adj. List
<b>Space</b>	<b>n+m</b>	<b>n<sup>2</sup></b>	<b>n+m</b>
<b>insertVertex</b>	<b>1</b>	<b>n</b>	<b>1</b>
<b>removeVertex</b>	<b>m</b>	<b>n</b>	<b>deg(v)</b>
<b>insertEdge</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>removeEdge</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>incidentEdges</b>	<b>m</b>	<b>n</b>	<b>deg(v)</b>
<b>areAdjacent</b>	<b>m</b>	<b>1</b>	<b>min( deg(v), deg(w) )</b>

**Q:** If we consider implementations of simple, connected graphs, what relationship between n and m?

- On connected graphs, is there one algorithm that underperforms the other two implementations?

**Q:** Is there clearly a single best implementation?

- Optimized for fast construction:

- Optimized for areAdjacent operations:

#### CS 225 – Things To Be Doing:

1. No lecture Friday Exam 4 CBTF Online
2. lab\_dict released this week; due on Sunday, Nov. 15
3. mp\_mazes final due date on Monday, Nov. 16
4. POTD today!