Graph Data Structure Implementation Worksheet

Name:_____

Assume all graphs have n vertices and m edges.

1. How much **space** is needed to store graphs using the two most common data structures? Fill out the following chart with as accurate of an answer as possible. In other words, go beyond just a Big-O estimate.

	Unweighted		Weighted	
	Undirected	Directed	Undirected	Directed
Adjacency Lists				
Adjacency Matrix				

2. Give Θ bounds for the **worst-case running time** of each of the following operations on an undirected graph using the two most common data structures. In each, `u` and `v` are vertices. In the two blank rows, list 2 more operations that might be helpful in comparing the two implementations of a graph and give their complexity.

	Adjacency List	Adjacency Matrix
`areAdjacent(u,v)`		
`degree(u)`		
Iterate over neighbors of `v`		
`addEdge(u,v)`		
`deleteEdge(u,v)`		

3. If you need to implement `insertVertex` and/or `deleteVertex`, would you be better off with an adjacency matrix or adjacency lists? Justify your choice using the appropriate criteria. (Note: *criteria* is plural.)

4. Under what conditions would you choose to use an **adjacency matrix** to represent a graph? Give specific criteria, using the information from the previous problems and considering the flavors of graphs discussed in the book and any other important factors that come to mind.

5. Under what conditions would you choose to use an **adjacency list** to represent a graph? Give specific criteria, using the information from the previous problems and considering the flavors of graphs discussed in the book and any other important factors that come to mind.