Graphs
- Nodes + edges = undirected graph.
- Nodes + arcs (= directed edges) = directed graph.
- Labels on nodes or arcs/edges are possible.

Paths
- In directed graph: sequence of nodes with arc from each node to the next.
- In undirected graph: sequence of nodes with an edge between each two consecutive nodes.
- Length of path = number of edges/arcs thereon.
- If edges/arcs are labeled by numbers, then we can sum the labels along a path to get a distance.

Cycles
- In directed graph: path that begins and ends at the same node.
  - Simple cycle: no repeats except the ends.
  - Note a cycle has many paths representing it, since the begin/end point may be any node on the cycle.
- In undirected graph: Simple cycle = sequence of 3 or more nodes with same begin/end point, but no other repetitions.
  - “Cycle” in undirected graph is tricky; see FCS, p. 456.

Adjacency List Representation
- Array or list of headers, one for each node.
- Undirected graph: header points to list of adjacent (shares an edge) nodes.
• Directed case: header for node \( v \) points to list of *successors* (nodes \( w \) with arc \( v \to w \)).
  - Predecessor = inverse of "successor."

• Labels for nodes may be attached to header for that node.

• Labels for arcs/edges are attached to the list cell for the "other" node.
  - Note an edge is represented twice.

**Adjacency Matrices**

• Node names must be integers \([0..\text{MAX}-1]\).

• \( M[i][j] = \text{TRUE} \) iff there is an edge between nodes \( i \) and \( j \) (arc \( i \to j \) in directed case).

• Node labels in separate array.

• Edge/arc labels can be the value of \( M[i][j] \).
  - Needs a special label that says "no edge/arc."

**Size Parameters**

We shall conventionally use:

• \( n \) = number of nodes of a graph.

• \( m \) = larger of number of nodes and edges/arc.
  - Note: \( m \geq n \).

**Class Problem**

A hub in an undirected graph is a node with an edge to every other node.

• How fast (as a function of \( n \) and \( m \)) can we find whether or not a graph has a hub?
  - If the graph is represented by adjacency lists? Adjacency matrix?
  - In the worst case? Average case? What is an "average case" anyway?