Java 2 Graphs

- FUCK GRAPHS

- **Graphs**
  - The most general data structure we've talked about.
  - Represents pairwise relationships between objects.
  - The objects are represented as vertices and the relationships are represented as edges between the vertices.
  - A graph is a **model** more than a collection.

- **Graphs: Motivating Problems**
  - The common thread is the usage of graphs as the model for representing this data.
- **Edge Characteristics**
  - Directed V. Undirected
    - Undirected edges represent symmetric relationships, and are indicated by a line.
    - Directed edges represent asymmetric relations, and are indicated by an arrow.
  - Weighted V. Unweighted
    - Edges can have numeric values (weights) associated with them or not. These values can represent cost, time, distance, etc.

- **Adjacency**
  - Adjacency is the basic connectedness property of vertices.
  - Undirected Case
    - Two vertices are **adjacent** to each iff there is an edge between them.
  - Directed Case
    - Node B is **adjacent** to node A iff there is an edge from A to B.

- **Adjacency Matrix**
  - An **adjacency matrix** is a two dimensional table where both the rows and the columns represent the vertices of the graph. Cell *(i, j)* indicates if vertex j is adjacent to vertex i.
  - An **adjacency list** is a one dimensional table where each entry represents the vertices of the graph. Entry k stores a linked list of all the vertices that are adjacent to vertex k.

- **Glossary**
  - **Self loop**
    - an edge that links a vertex to itself
  - **Simple graph**
    - a graph with no self loops
  - **Path**
    - A sequence of vertices/edges from a start vertex to an end vertex.
  - **Simple Path**
    - A path that does cross the same edge twice.
  - **Cycle**
    - A simple path that starts and ends at the same vertex.
  - **Acyclic Graph**
    - A graph with no cycles.
  - **Connected Graph**
    - A graph in which there is a simple path between any two pair of vertices.
  - **Connected Component**
    - Maximal subgraph that is connected.
  - **Complete Graph**
    - A simple graph in which every pair of vertices is adjacent to each other.
  - **Spanning Tree**
    - A spanning tree of a connected, undirected graph is a connected, acyclic subgraph that contains

all the vertices of the graph.

- **Depth-First**
    - Explore the graph by looking for new vertices far away from the start vertex, and examining nearer vertices only when dead ends are encountered.
    - Will visit each vertex that is reachable from the start vertex.

- **Breadth-First**
    - Explore the graph by looking at all the vertices closest to the start vertex, and move farther away only when everything has been examined.

- **Take-Away**
    - BFS and DFS can be modified to add a lot more functionality to their individual search schemes.