**Graphs**

* A graph G is an ordered pair of a set V of vertices and a set E of edges. G = (V, E). Essentially, you write vertices first, and edges second.
* Vertices are connected together by a set of edges
* **Directed vs Undirected Graph**– Direction implies that the connection goes one way, and undirected means both ways. Directed is similar to trees.
* Say there are two vertices, u and v, with a directed edge from u to v. You write this as (u,v), since the directional edge means order matters. If the edge is undirected, write it as {u,v}.
* **Weighted vs Unweighted** – The edges have weights associated with them. In other words, each edge is not equal. If you go from point A to point B in a city, you’ll have many roads associated with this path. However, certain paths are more optimal, since they take lesser time. So, weights should be used to represent more attractive edges you want to take to traverse through this graph.

**Properties**

* |V| is number of vertices in the graph, and |E| is the number of edges.
* **Self – loop** – The edge connects the same vertex to itself.
* **Multiedge** – You have multiple edges connecting two vertices. Think of flights between two cities, but multiple flights go back and forth. Each edge might represent prices.
* **Simple graph** – Graph doesn’t contain Multiedge or self-loop.
* If there are n vertices, then max number of edges is n(n – 1) if directed.
* The max for undirected is half of directed, since two nodes can only be connected by one edge.
* A dense graph has many edges, and sparse not too many.
* **Path** – A sequence of vertices where each adjacent pair is connected by an edge.
* **Simple Path** – A path in which no vertices (and thus no edges) are repeated.
* **Trail** – A path in which no edges are repeated.
* **Strongly connected graph** – If there is a path from any vertex to any other vertex.
* **Closed walk** – Starts and ends at same vertex.
* **Cycle and Acyclic graph**