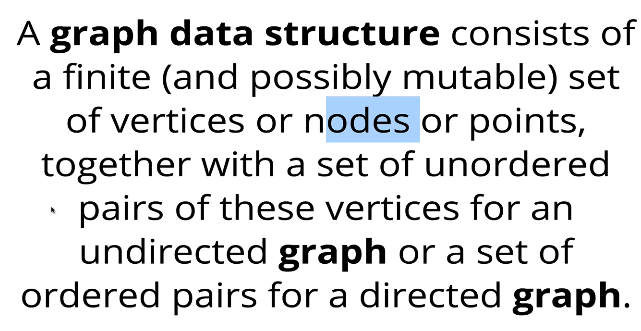
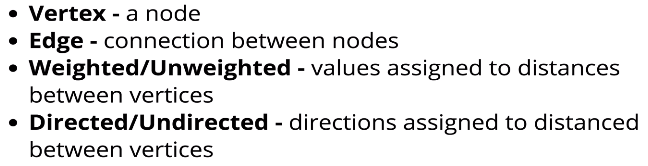
JavaScript Algorithms and Data Structures Masterclass

# Section 26: Graphs

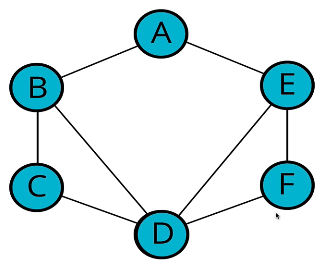
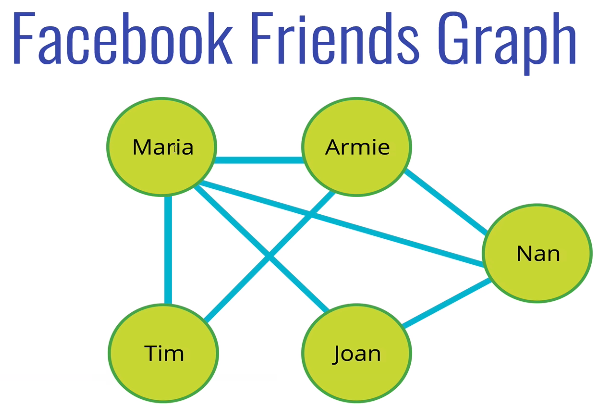
## Intro to Graphs

* 
* It is a DS that consists a set of Nodes and a set of pairs (Nodes + Connections)
  + Nodes connected to each other
* **Examples**
  + Social Networks
  + Location/google maps
  + Recommendation Engines (**ex**. Netflix)
  + Routing algos
  + Visual Hierarchy
  + File Systems

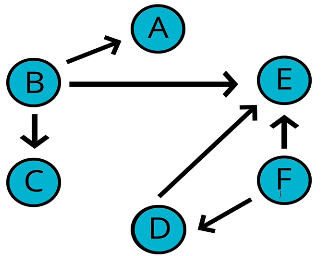
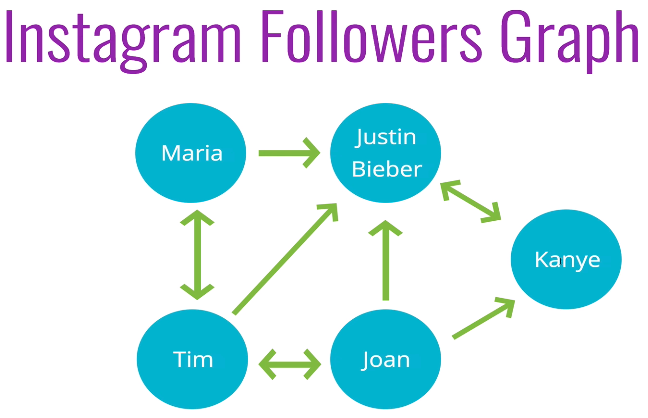
## Types of Graphs

* **Terms**:
  + 

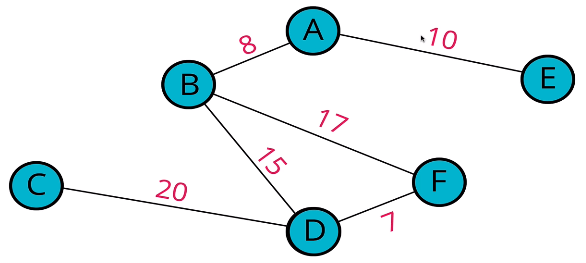
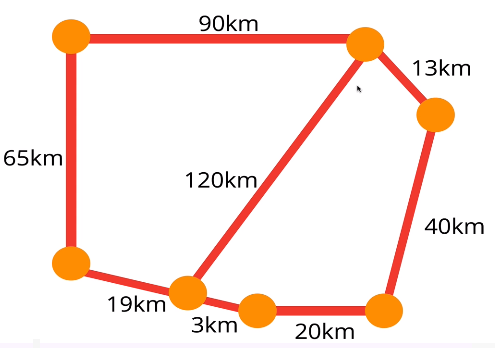
### Undirected Graph:

* **General Diagram**:
  + 
* **Real Example**:
  + 

### Directed Graph

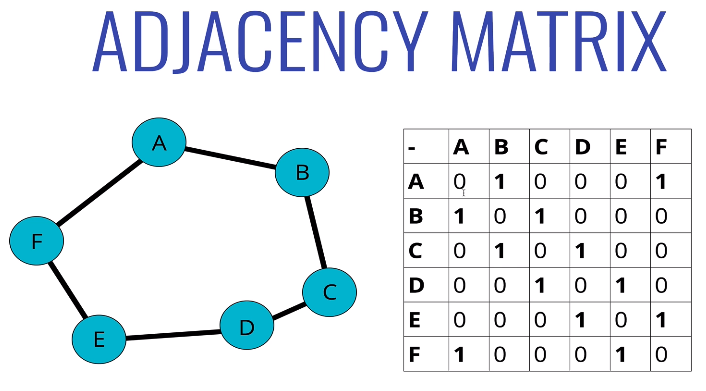
* **General Diagram**
  + 
* **Real Example**
  + 

### Weighted Graph

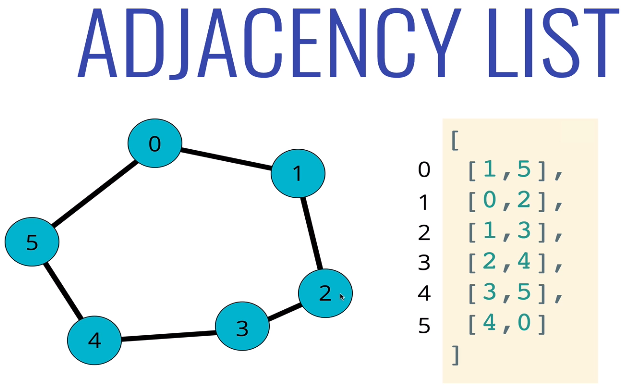
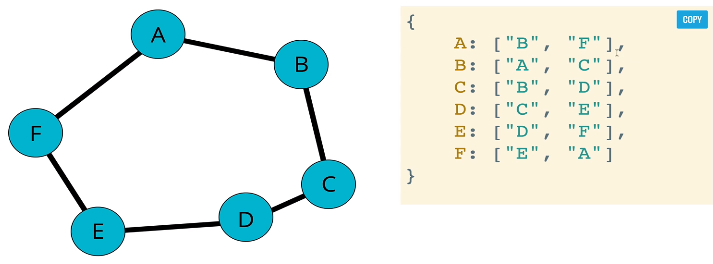
* The edges have a value assigned to them
  + **General Example**
    - 
  + **Real Example**
    - 

## Storing Graphs

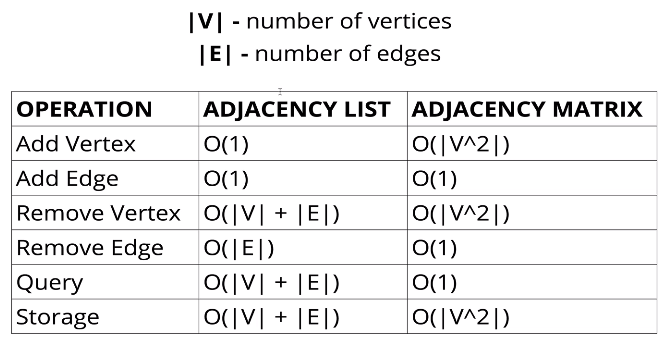
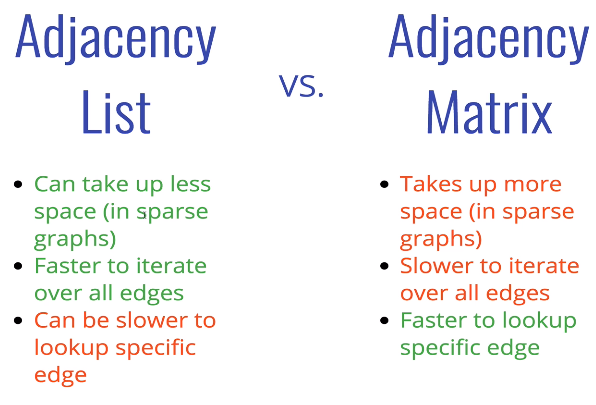
### Adjacency Matrix

* **Diagram**
  + 
    - Boolean represents whether a edge(s) (0 or 1) exists between vertex (nodes)

### Adjacency List

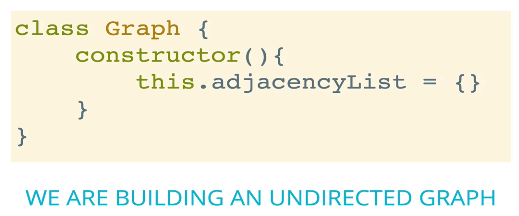
* **Diagram (Array)**
  + 
* **Diagram (Hash Table)**
  + 

## Big O of Adjacency List vs Adjacency Matrix

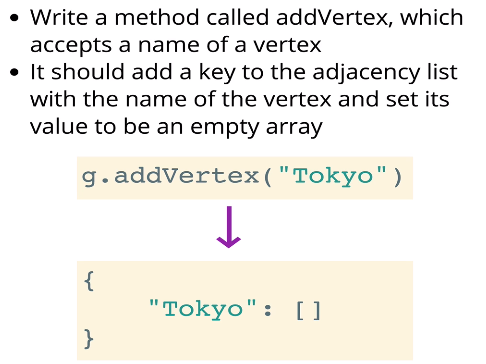
* 
* **Comparison**
  + 

## Implementation

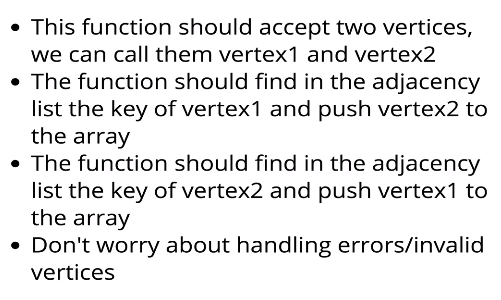
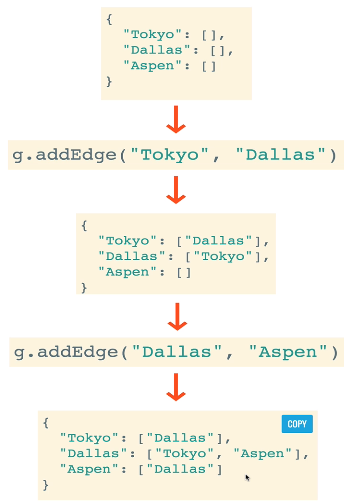
### Graph Class

* 

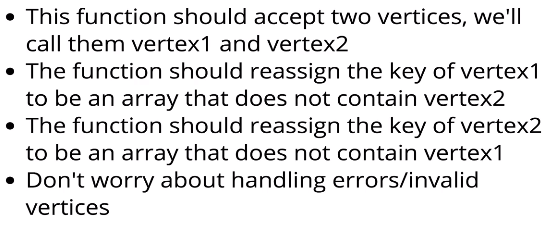
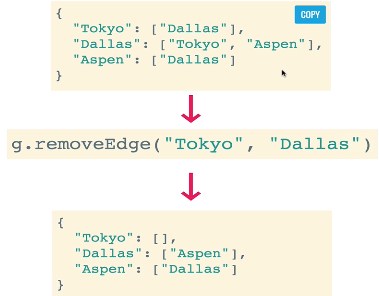
### AddVertex()

* **Psuedo-code**:
  + 

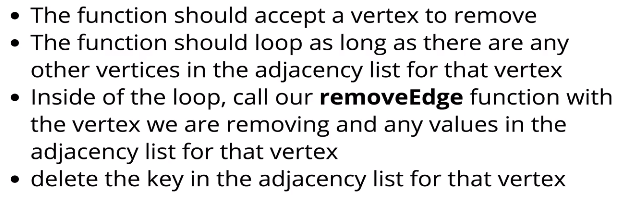
### AddEdge()

* **Psuedo-code:**
  + 
    - **Examples**
      * 

### RemoveEdge()

* **Psuedo-code**:
  + 
  + 

### RemoveVertex()

* **Psuedo-code**:
  + 
  + 