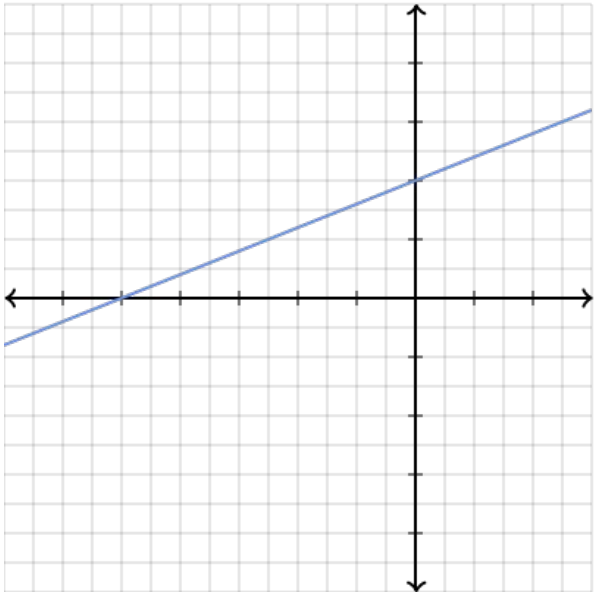
# Graph Theory I

By: Ethan Pronev

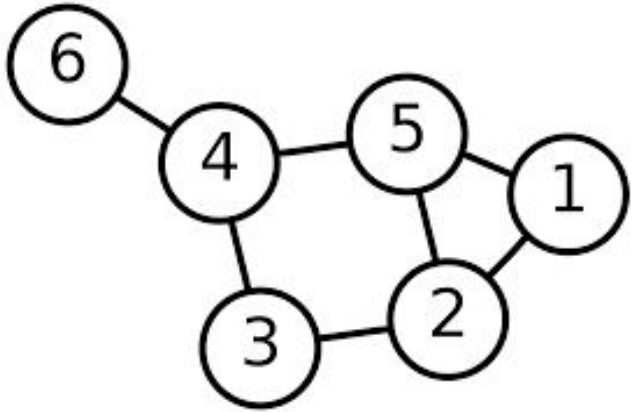
# What is Graph Theory?

Not this kind of graph



In computer science, a graph is a data structure that consists of vertices (also called nodes) and edges

Vertices are distinct points, and edges create links between multiple points



# Relevant Terminology

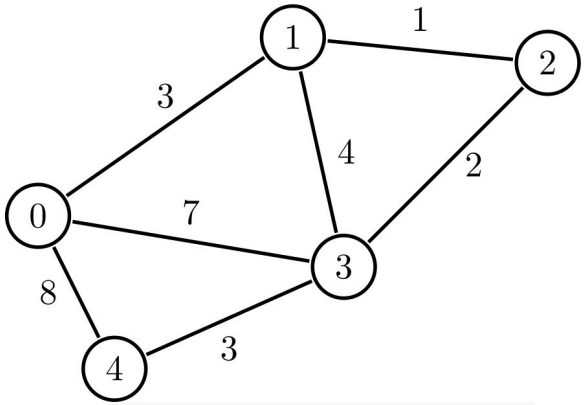
Adjacent - two vertices are adjacent if there is an edge between them

Degree - the degree of a vertex is the number of nodes adjacent to it

Cycle - a path on a graph from a vertex to itself

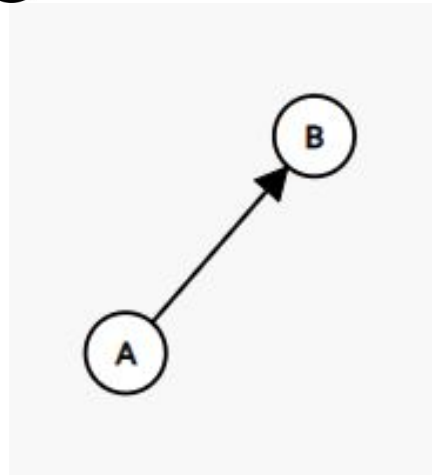
# Types of Graphs

Unweighted/Weighted - Each edge in a graph can have a ‘weight’ associated with it (in different problems this could represent travel time, distance, cost, etc.)



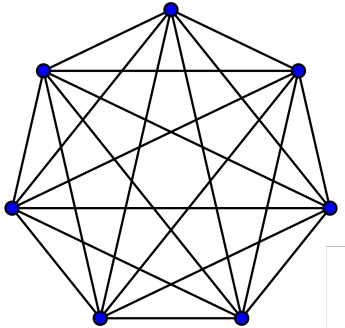

Directed/Undirected - In some graphs the edges may only be traversed in one direction

- Eg. B is connected to A, but A is not connected to B

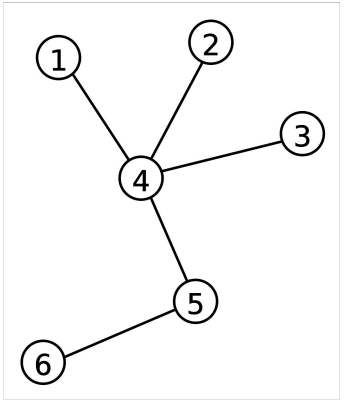


# Types of Graphs

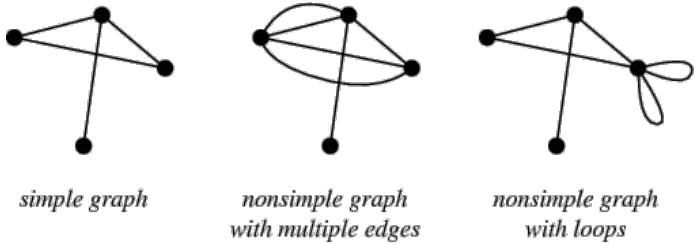
Complete - Every node is adjacent to every other one



Tree - no cycles of edges and nodes (contains n vertices and n-1 edges)




Simple - Simple graphs may not contain vertices adjacent to themselves or multiple edges between the same pair of vertices



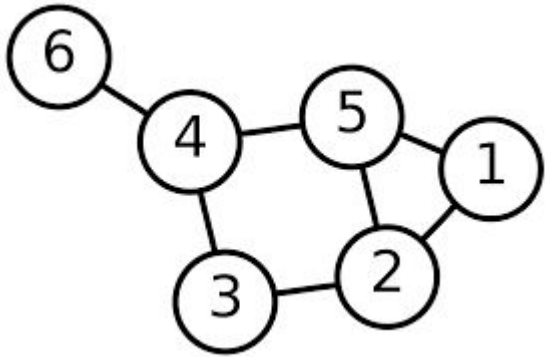
# How to Represent Graphs in Code

Method 1: Edge List Representation

Essentially a list of every edge in the graph in the form {a,b} where a and b are every pair of nodes that share an edge

Example:

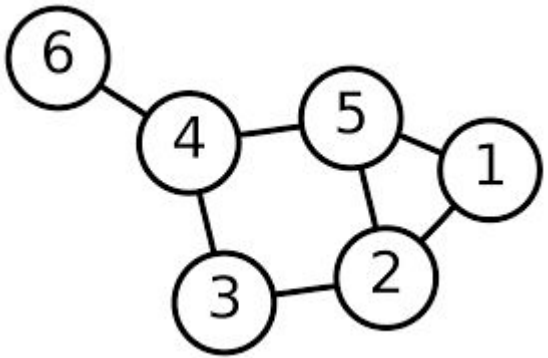
E = {{6,4},{4,5},{5,2},{2,3},{3,4},{5,1},{1,2}}



# How to Represent Graphs in Code

Method 2: Adjacency Matrix

For n nodes, you would use a n*n 2D array where array[i][j]=true if i and j are adjacent, and array[i][j]=false if they are not



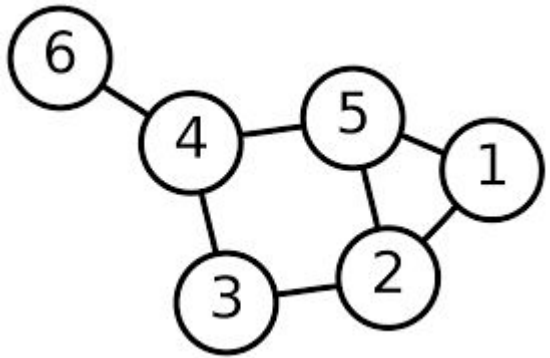

| | **1** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|---|
| **1** | 0 | 1 | 0 | 0 | 1 | 0 |
| **2** | 1 | 0 | 1 | 0 | 1 | 0 |
| **3** | 0 | 1 | 0 | 1 | 0 | 0 |
| **4** | 0 | 0 | 1 | 0 | 1 | 1 |
| **5** | 1 | 1 | 0 | 1 | 0 | 0 |
| **6** | 0 | 0 | 0 | 1 | 0 | 0 |

# How to Represent Graphs in Code

Method 3: Adjacency List

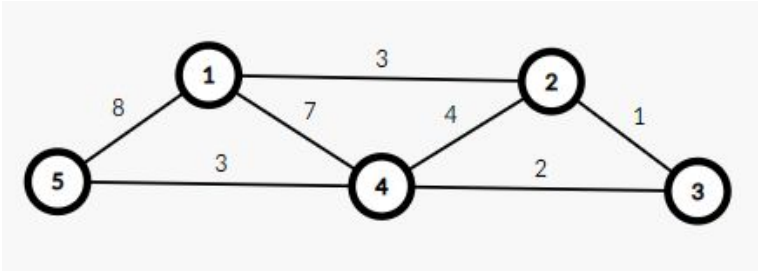
An array of vectors is used

array[i] = {a,b,c...} indicates that node i is adjacent to a, b, c...




| | |
|---|---|
| **1** | { 2 , 5 } |
| **2** | { 1 , 3 , 5 } |
| **3** | { 2 , 4 } |
| **4** | { 3 , 5 , 6 } |
| **5** | { 1 , 2 , 4 } |
| **6** | { 4 } |

# Graph Representation - Weighted Graphs




Edge List Representation

Use triplets instead of pairs to store each edge

Form is {a,b,w} where a and b are the nodes and w is the weight

Example: