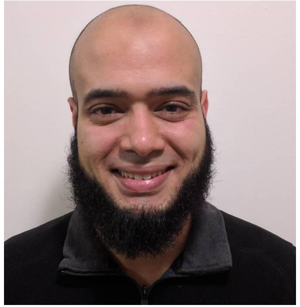
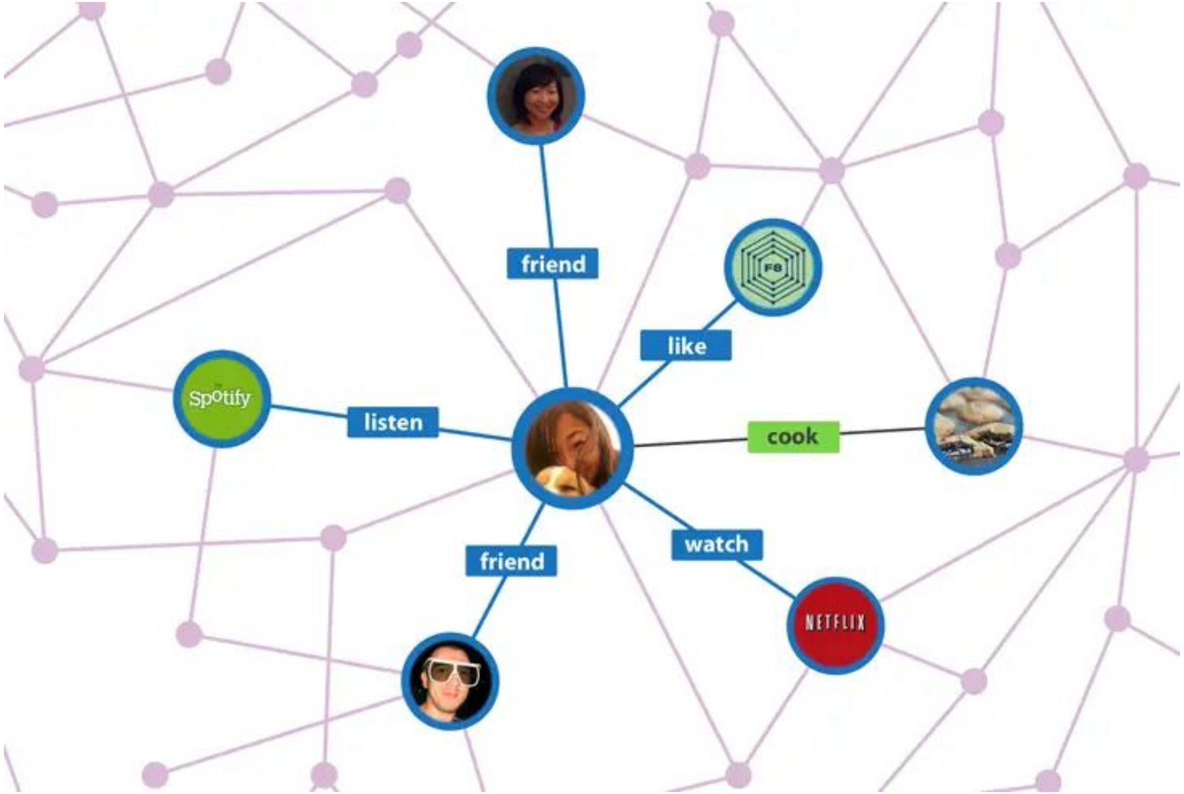
Machine Learning
Graph Theory

Mostafa S. Ibrahim
Teaching, Training and Coaching for more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / MSc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



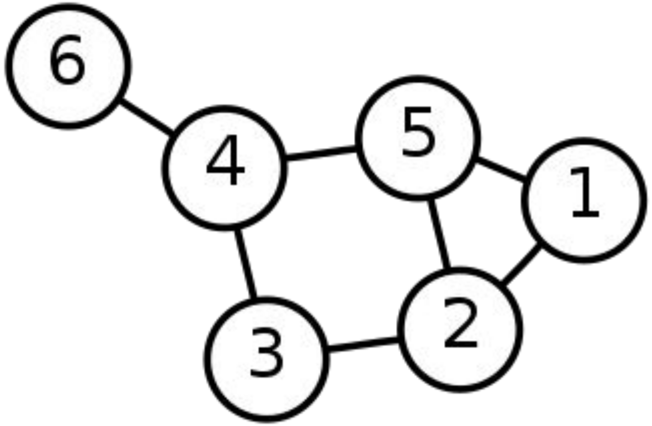
Social Media: people relationships



Img src

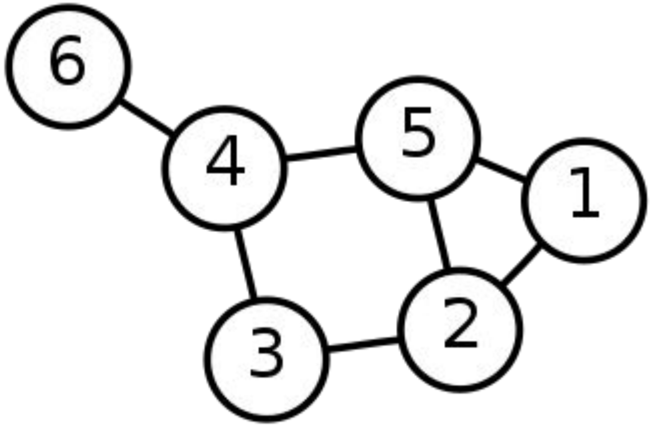
Graph Theory

- There are many things that involve **pairwise relations** between **objects**
- Think about a city of 6 internal airports and flights between them
- Let's number the airports from 1 to 6
- The diagram below is called a **graph**
  - We use it to model the relationships
- It tells us
  - There is a connection from 1 to 5
  - There is a connection from 1 to 2
  - But there are no further connections from 1



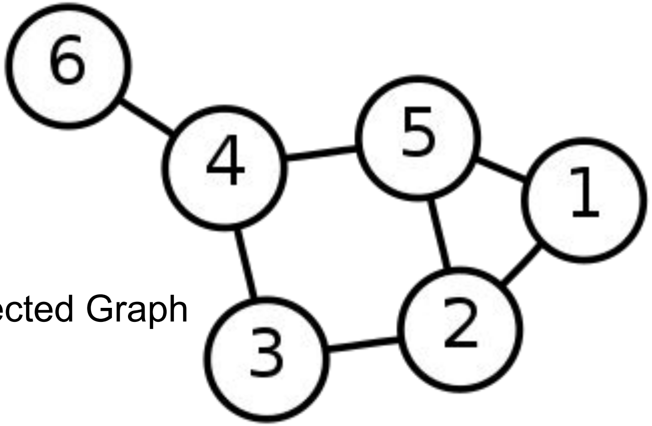
Graph

- A graph is a structure consisting of vertices V and edges E:
- V: Collection of vertices (or **nodes**)
  - V = {1, 2, 3, 4, 5, 6}
- E: Collection of edges
  - Edge = a link between a pair of vertices
  - E = {(1,2), (1,5), (2, 3), (2, 5), (3, 4), (5, 4), {4, 6}}
- Source and Target
  - Given an edge from A to B
  - We may call A the source, and B the target

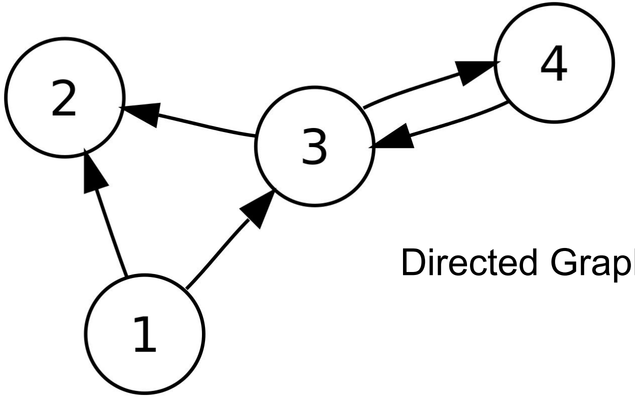


Directed vs Undirected Graphs

- Undirected graphs: edges don’t have a direction (two-way)
  - FB Example: Mostafa is a friend to Belal, and Belal is a friend to Mostafa
- Directed graphs: edges with specific one-way direction
  - FB Example: Mostafa follows Ziad (then Mostafa sees Ziad’s posts), but not the opposite
    - We can still have ANOTHER directed edge from Ziad to Mostafa



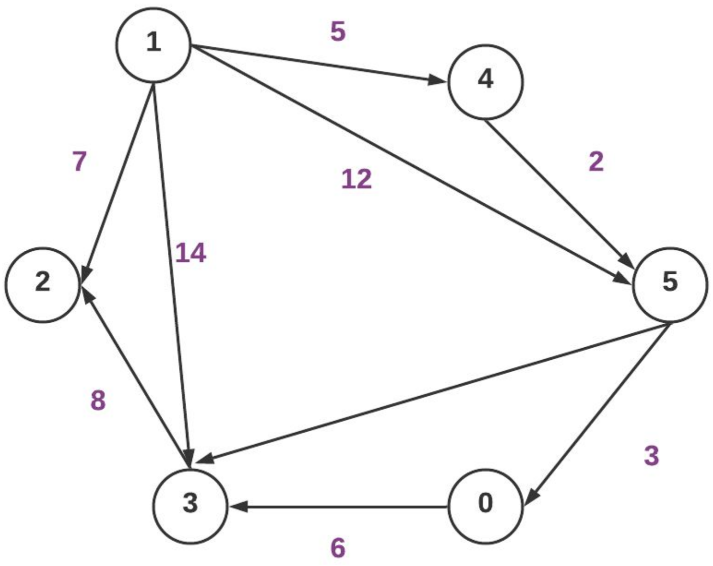
Undirected Graph



Directed Graph

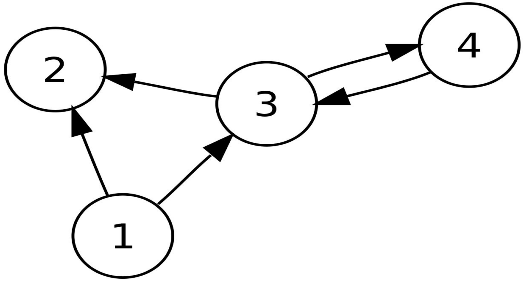
Weighted Graphs

- Weight is a **value** for the **edge**.
- To model airports, each flight needs a weight (e.g. trip length or cost)



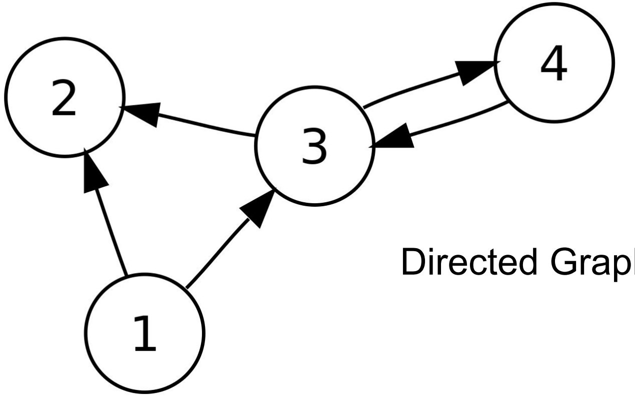
Graph Cycle

- A cycle is a **path** that starts and ends at the **SAME node**
  - In terms of definitions: usually *no other nodes are repeated*
- [3, 4, 3] is the **only** cycle.
- [1, 2, 3] ia NOT a cycle
- A directed acyclic graph (**DAG**): a directed graph with no **directed** cycles



Node neighbours

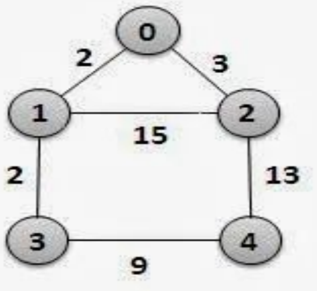
- Node u has node v as a neighbour IFF (u, v) is an edge
  - node 2 has no neighbours!
  - Meanwhile, node 1 has 2 neighbours {2, 3}



Directed Graph

Adjacency Matrix Representation

- Representing a graph of V nodes using a 2D matrix VxV
  - Index the vertices: {0, 1, 2, 3…., V-1}
- mat[i][j] = **edges weight** between node(i) and node(j)
- There is another dynamic representation called adjacency list



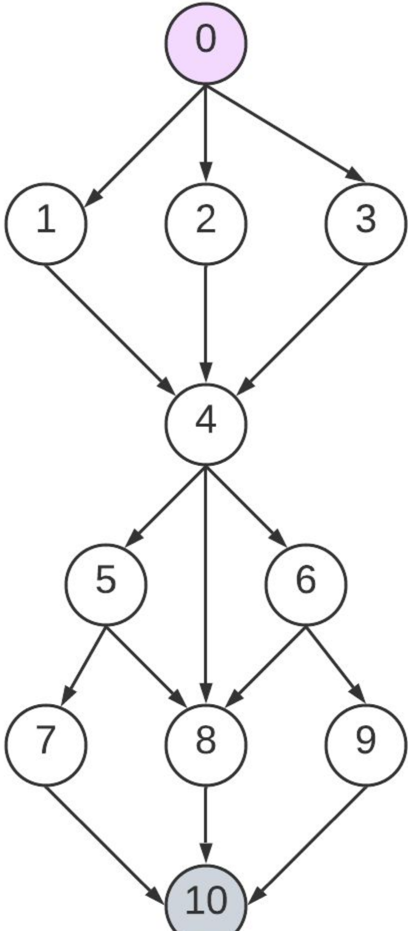
	0	1	2	3	4
0	0	2	3	0	0
1	2	0	15	2	0
2	3	15	0	0	13
3	0	2	0	0	9
4	0	0	13	9	0

In Neural Network

- We care about weighted directed **DAG** graphs
- We use adjacency matrices for representation
  - Later, we multiply these matrices, which can be so fast using GPUs
- In terms of brains:
  - Nodes are called Neurons
  - Edges are called synapses

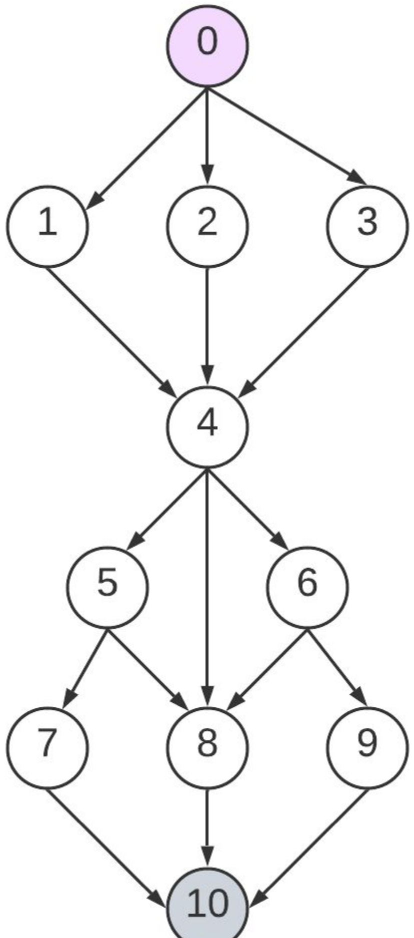
Challenge: Number of paths in DAG

- Given a Directed Acyclic Graph, find the number of simple paths between 2 given nodes in linear time
- How many paths between:
- A: 10 and 10?
- B: 9 and 10?
- C: 6 and 10?
- D: 4 and 10?
- E: 2 and 10?
- F: 0 and 10?
- What is one popular algorithm behind such problems on DAG?



Challenge: Number of paths in DAG

- This is a classical dynamic programming (DP) problem!
  - Recursive solutions that can be saved to avoid duplications!
- DAG is always a classical application for DP
  - Because node like 4 is shared - avoid duplications
- Let F(node) = # of paths from node to 10
  - **F(10)** = 1   base case
  - **F(7)** = F(8) = F(9) = 1
  - **F(5)** = F(6) = 2
  - **F(4)** = F(5) + F(6) + F(8) = 2 + 2 + 1= 5
  - **F(1)** = F(2) = F(3) = F(4) = 5
  - **F(0)** = F(1) + F(2) + F(3) = 5+5+ = 15
- We either implement it recursively (top-down) or iteratively (bottom-up), like we did
- This is a key property in backpropagation algorithms!



“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”