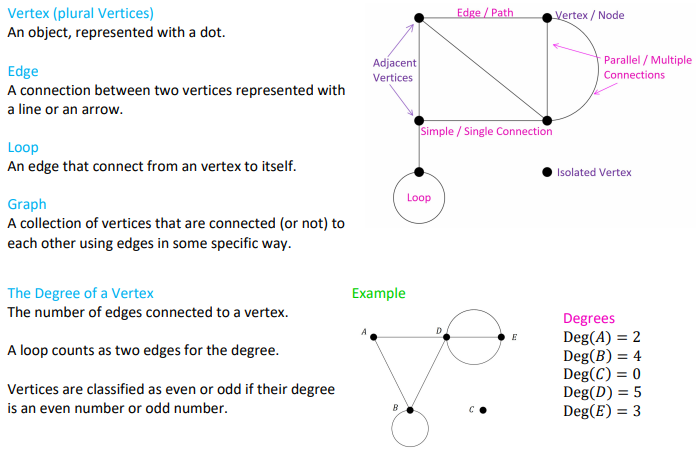
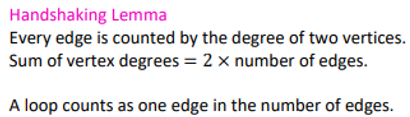
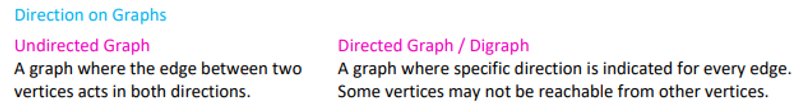
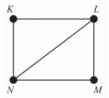
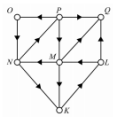
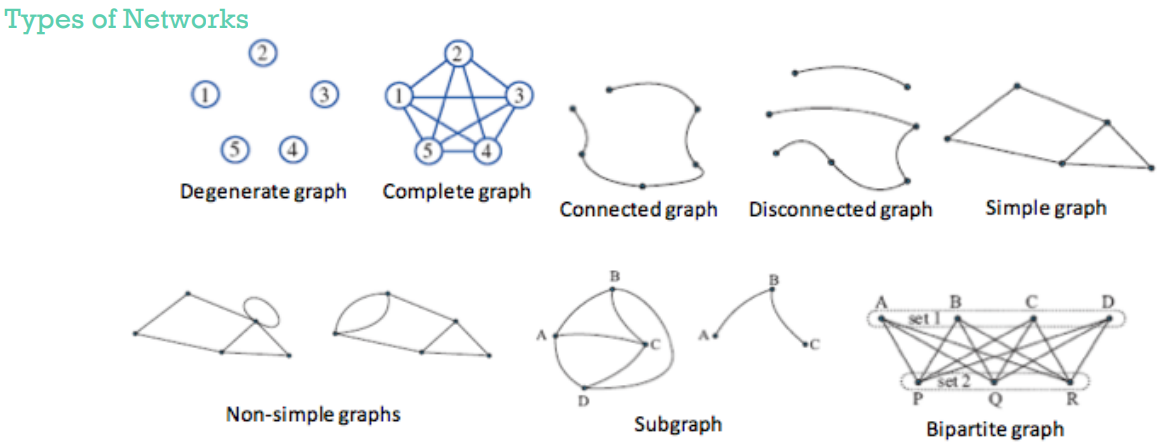
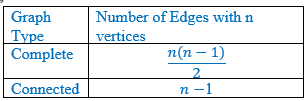
Networks

**** 



**Undirected Graphs**

## ***Types of graphs***

**Simple graph** – No loops or duplicate edges.

**Isolated vertex** – A graph has an isolated vertex if there is a vertex that is not connected to another vertex by an edge.

**Degenerate graph** – Degenerate graphs have all vertices isolated. Therefore, there are no edges in the graph at all.

**Connected graph** – Each vertex is either directly or indirectly connected to every other vertex.

**Bridge** – A bridge is an edge that when removed makes the graph unconnected.

**Subgraph** – Are graphs that are part of larger graphs.

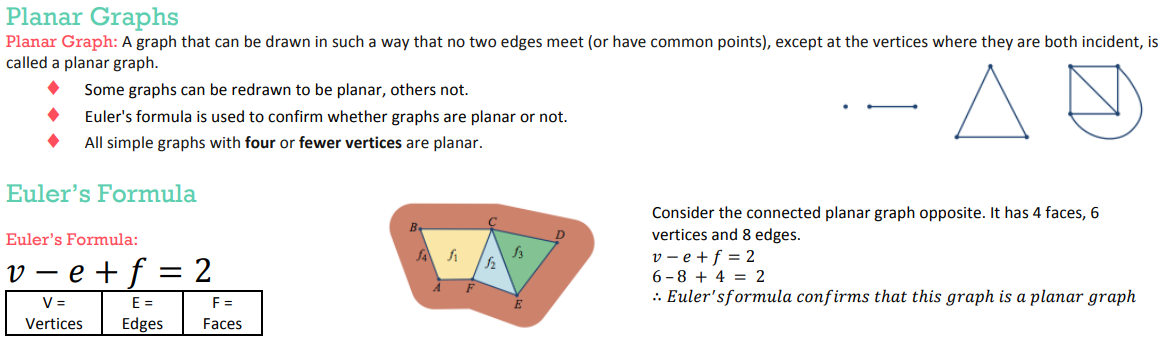
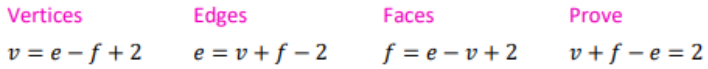
**Equivalent (isomorphic) graph** – Look different but have the same information

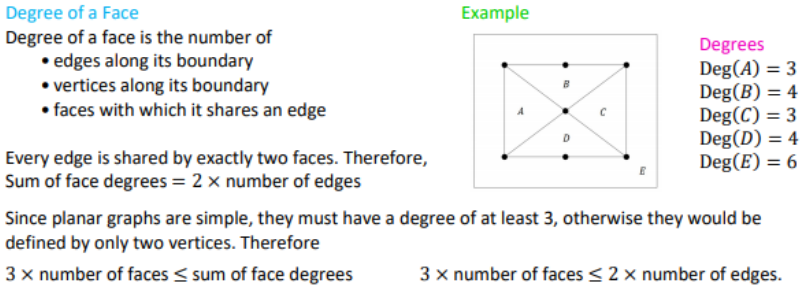
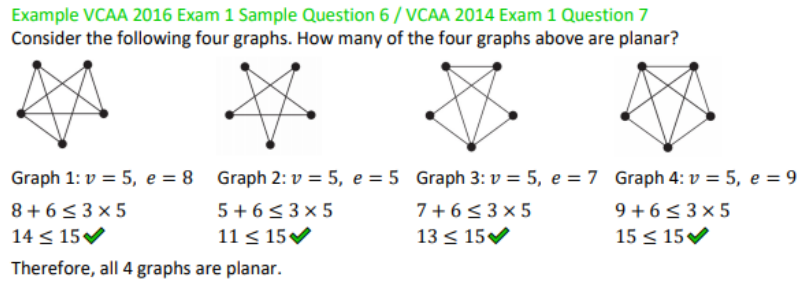
**Complete graph** – Every vertex has a direct connection to every other vertex.

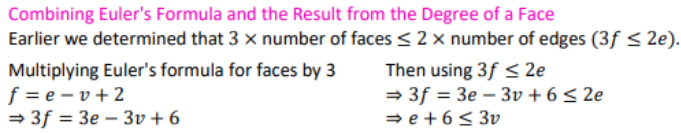
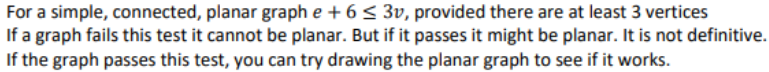
**Bipartite Graph** – A bipartite graph is a graph whose set of vertices can be split into two subsets X and Y in such a way that each edge of the graph joins a vertex in X and a vertex in Y.

**Isomorphic graphs** –Two graphs have: ① same numbers of edges and vertices; ② corresponding vertices have the same degree and the edges connect the same vertices.

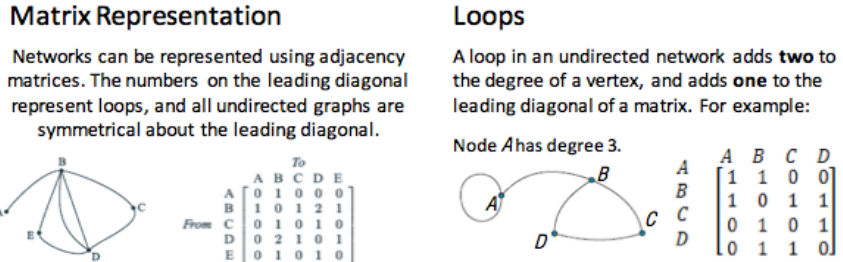
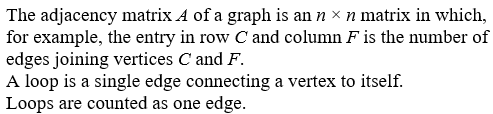
## ***Planar Graphs & Euler’s Formula***

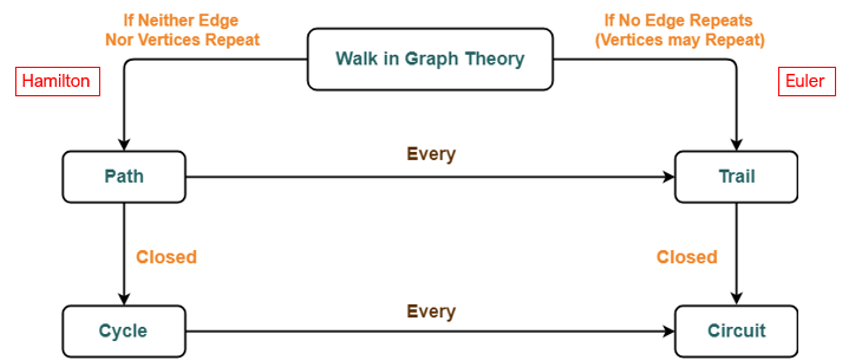
 

## ***Adjacency Matrix Representation***

## ***Euler & Hamilton***



**Travelling in graphs**

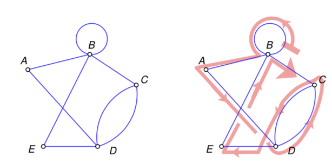
**Route** – A description of your travels, given by the vertices visited in the order they are visited.

**Walk** – A walk can be any type of journey within a graph, you can walk wherever you wish.

**Trail** – A special kind of walk, you **can’t repeat** any of the **edges** that you have taken, but you can revisit vertices.

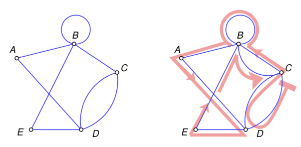
**Path** – A path is a special kind of trail, with a path you **can’t repeat** any **edges or vertices**.

**Eulerian trails and circuits**

**Eulerian trails** – Is a trail in which every **edge** is visited **once**. Vertices can be repeated.

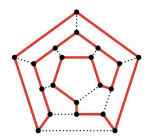
A Eulerian trail will only exist if:

* The graph is connected
* The graph has exactly two vertices of an odd degree

**Eulerian circuit** – Is a Eulerian trail (travels every **edge once**) that begins and ends from the same vertex.

A Eulerian circuit will only exist if:

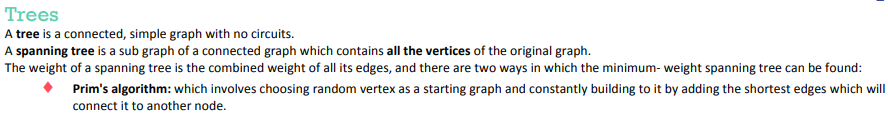
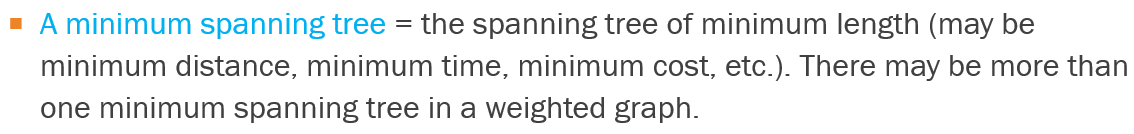
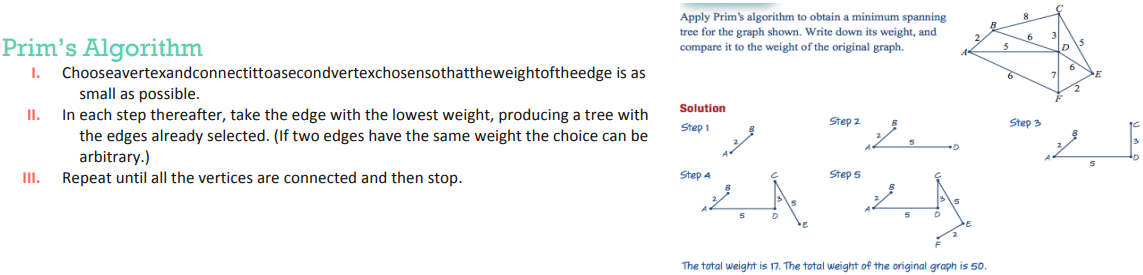
* The graph is connected
* All the vertices have an even degree

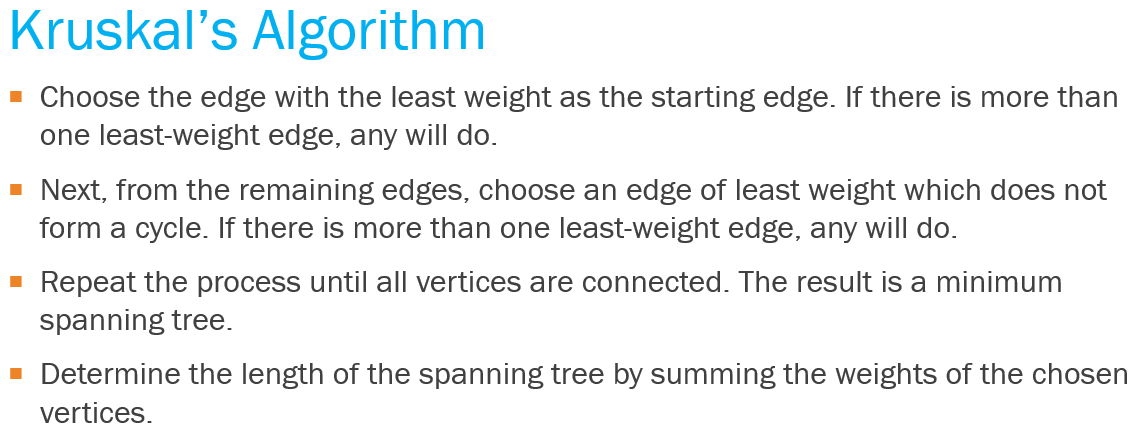
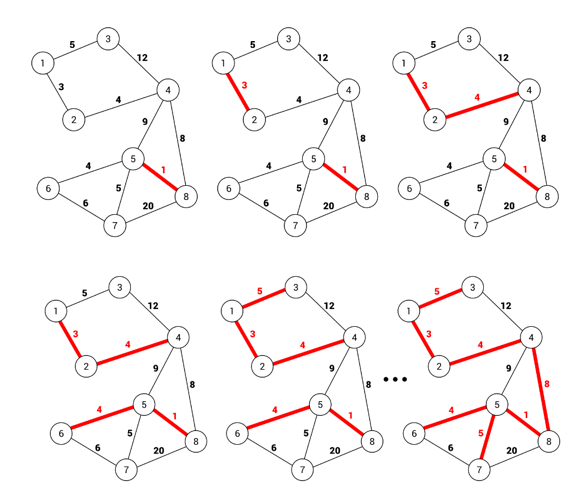
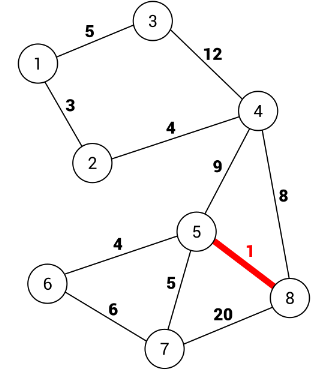
**Hamiltonian paths and cycles**

**Hamiltonian path** – Is a path that visits all of the **vertices** in a graph **only once**.

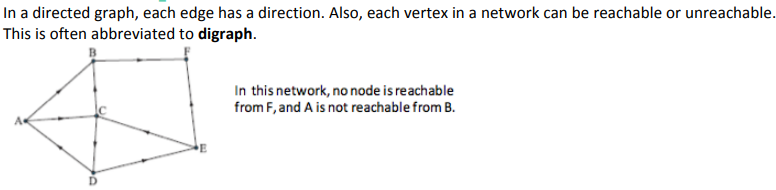
**Hamiltonian cycle** – Is a cycle that visits every vertex and begins and ends at the same vertex.

## ***Weighted Graphs***

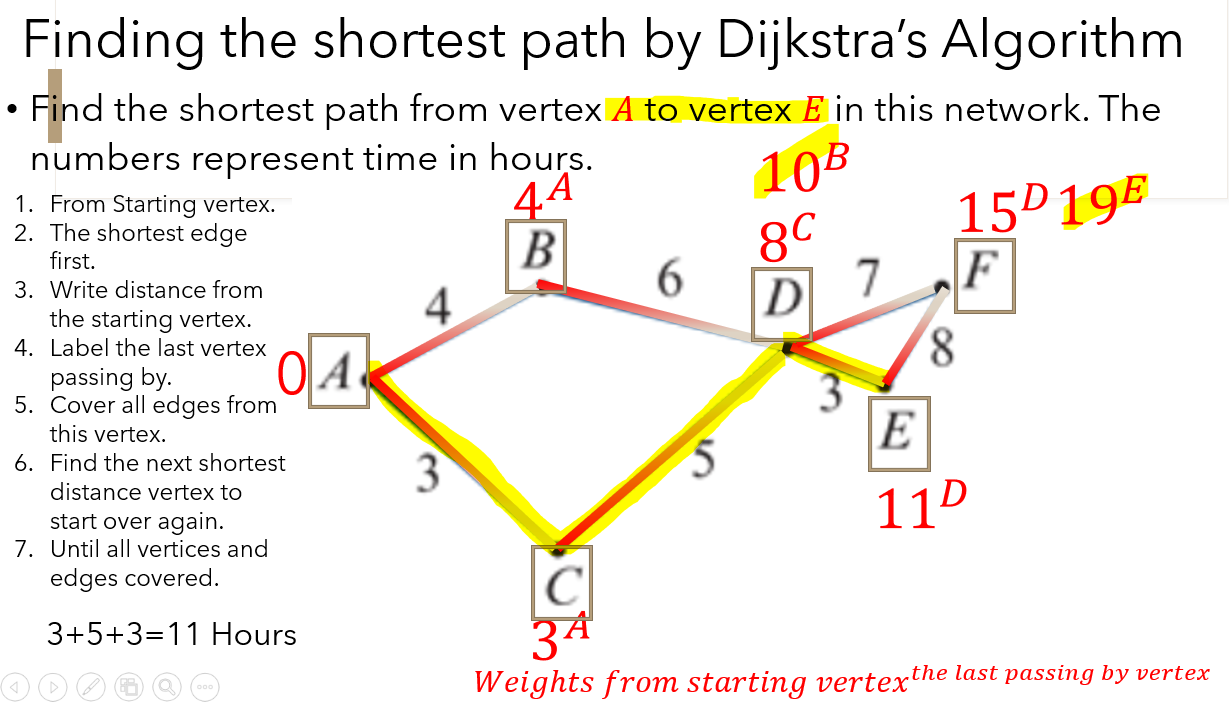
 

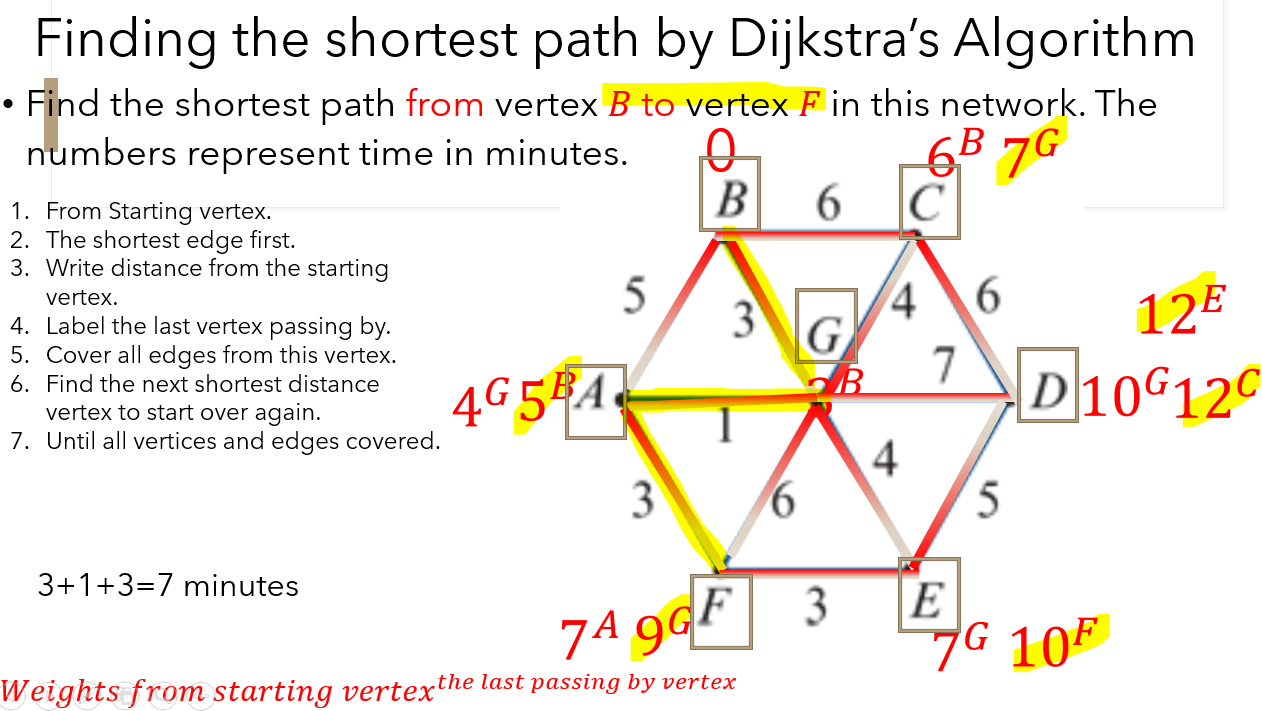
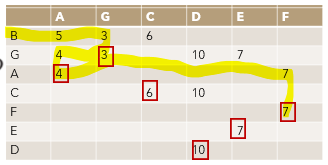
  

**Directed Graphs**



## ***Shortest Path Problem***

****

****

## ***Different Types of Greedy Algorithm***

Prim's Minimal Spanning Tree Algorithm

Kruskal’s Minimal Spanning Tree algorithm

Dijkstra's Shortest Path Algorithm

Ford-Fulkerson Networks Flows Algorithm

## ***Mathematical Terminologies***

|  |  |  |  |
| --- | --- | --- | --- |
| **Undirected Graphs** | | **Directed Graphs** | |
| **Terminologies** | **Algorithm** | **Terminologies** | **Algorithm** |
| Eulerian trails | Exactly 2 vertices of an odd degree | The Maximum Flow | Ford-Fulkerson Algorithm |
| Eulerian circuits | All vertices even degree | The Shortest Path | Dijkstra's Algorithm |
| Hamiltonian paths | Visits all of the vertices in a graph only once | Matching & Allocation Problems | Hungarian Algorithm |
| Hamiltonian cycles | Visit All vertices, begin & end @ the same vertex | Critical Path Problems | Forward scanning = Biggest Number  Backward scanning = Smallest Number  Float = LST―EST |
| Minimal Spanning Tree | Prim's Algorithm, Kruskal’s Algorithm |