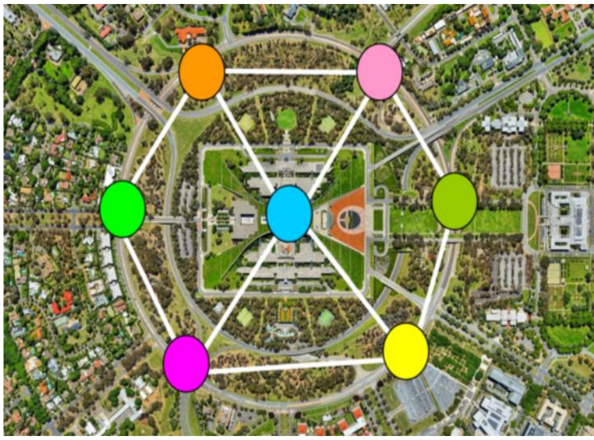
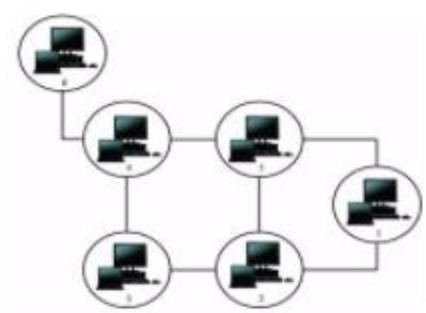




Chapter 8
Graph Data Structure

By: Dr. Aryaf A. Al-adwan
Faculty of Engineering Technology
Computer and Networks Engineering Dept.

Data Structures Course

Graph Data

Structure

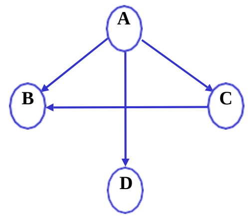
Outlines: Graphs Part-1

- Graph Terminology
  - Undirected vs. Directed
  - Adjacent Vertices
  - Path & Simple Path
  - Loops
  - Cycles & Simple Cycles
  - Weighted Graph
  - Complete Graph
  - Subgraph
  - Connected Graph & Component
  - Strongly vs. Weakly Connected
  - Symmetric Graph
  - Tree vs. Forest
  - End vertices (End points)
  - Parallel edges
  - Number of Edges
  - Sum of Vertex Degrees
  - In-degree and Out-degree
- Applications on Graphs
  - Communication Network
  - Driving Distance /Time Map
  - Street Map
  - Computer Networks
- Graph Representation
  - Adjacency Matrix
  - Adjacency Lists

Graph Terminology

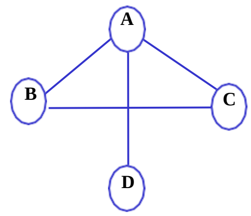
- A graph $G = (V, E)$
  - V = set of vertices (nodes or points)
  - E = set of edges (arcs or lines) = subset of $V \times V$
  - Thus $|E| \leq |V|^2 = O(|V|^2)$
- In an *undirected graph:*
  - $edge(u, v) = edge(v, u)$
- In a *directed* graph:
  - $edge(u,v)$ goes from vertex u to vertex v, notated $u \rightarrow v$
  - $edge(u, v)$ is not the same as $edge(v, u)$

Graph Terminology



Directed graph:

$V = \{A, B, C, D\}$
$E = \{(A,B), (A,C), (A,D), (C,B)\}$

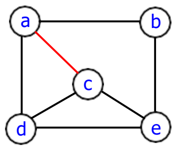


Undirected graph:

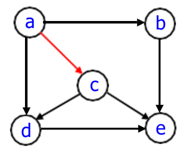
$V = \{A, B, C, D\}$
$E = \{(A,B), (A,C), (A,D), (C,B), (B,A), (C,A), (D,A), (B,C)\}$

Graph Terminology

- ***Adjacent vertices***: connected by an edge
  - Vertex ***v*** is adjacent to ***u*** if and only if **(*u*, *v*)** $\in$ ***E***.
  - In an undirected graph with edge **(*u*, *v*)**, and hence **(*v*, *u*)**, ***v*** is adjacent to ***u*** and ***u*** is adjacent to ***v***.



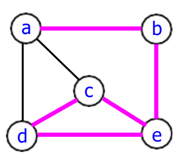
Vertex *a* is adjacent to *c* and vertex *c* is adjacent to *a*



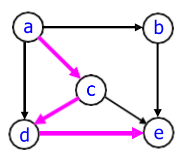
Vertex *c* is adjacent to *a*, but vertex *a* is NOT adjacent to *c*

Graph Terminology

- ***A Path*** in a graph from u to v is a sequence of edges between vertices $w_0, w_1, \ldots, w_k$, such that $(w_i, w_{i+1}) \in E$, $u = w_0$ and $v = w_k$, for $0 \le i < k$
  - The length of the path is k, the number of edges on the path



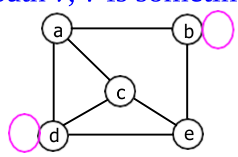
abedce is a path.
cdeb is a path.
bca is NOT a path.



acde is a path.
abec is NOT a path.

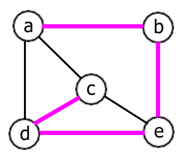
Graph Terminology

- ***Loops***
  - If the graph contains an edge (***v***, ***v***) from a vertex to itself, then the path ***v***, ***v*** is sometimes referred to as a ***loop***.



  - The graphs we will consider will generally be loopless.

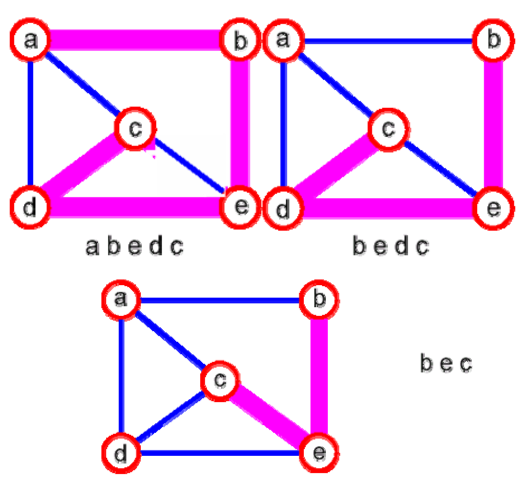
- ***A simple*** path is a path such that ***all vertices are distinct***, except that the first and last could be the same.



abedc is a simple path.
cdec is a simple path.
abedce is NOT a simple path.

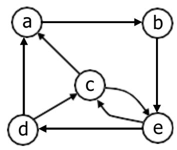
Graph Terminology

- simple path: no repeated vertices



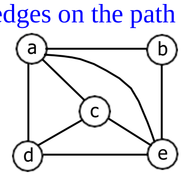
Graph Terminology

- ***Cycles***
  - A ***cycle*** in a **directed graph** is a **path** of length at **least 2** such that the **first** vertex on the path is the same as the **last** one; if the path is **simple**, then the cycle is a ***simple cycle***.



abeda is a simple cycle.
abeceda is a cycle, but is NOT a simple cycle.
abedc is NOT a cycle.

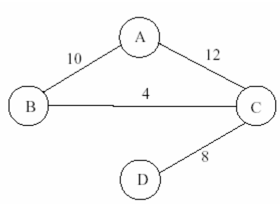
  - A ***cycle*** in a undirected graph
    - A path of length at **least 3** such that the **first** vertex on the path is the same as the **last** one.
    - The edges on the path are **distinct**.



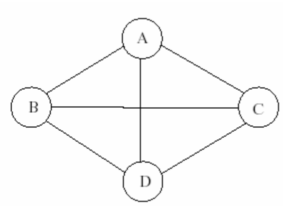
aba is NOT a cycle.
abedceda is NOT a cycle.
abedcea is a cycle, but NOT simple.
abea is a simple cycle.

Graph Terminology

- If each edge in the graph carries a value, then the graph is called ***weighted graph***.
  - A weighted graph is a graph $G = (V, E, W)$, where each edge, $e \in E$ is assigned a real valued weight, $W(e)$.
- A ***complete graph*** is a graph with an edge between every pair of vertices.
  - A graph is called ***complete graph*** if every vertex is adjacent to every other vertex.



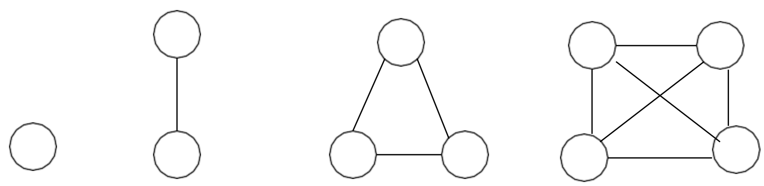
weighted graph



complete graph

Graph Terminology

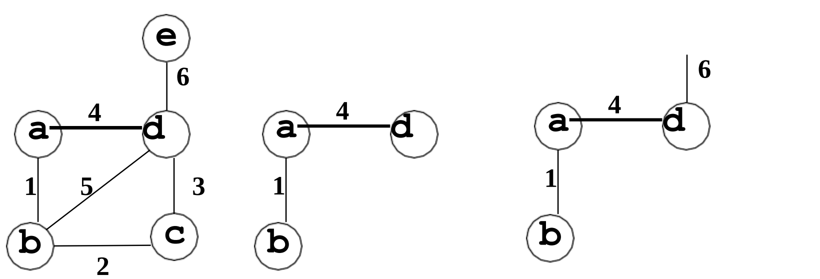
- Complete Undirected Graph
  - has all possible edges



$n = 1$ $n = 2$ $n = 3$ $n = 4$

Graph Terminology

- subgraph: subset of vertices and edges forming a graph
  - A graph $G_s = (V_s, E_s)$ is a ***subgraph*** of a graph $G = (V, E)$ if $V_s \subseteq V$, $E_s \subseteq E$, and $E_s \subseteq V_s \times V_s$.



G

is a sub-graph

Not a sub-graph

since, $E_s = \{1,4,6\} \not\subset V_s \times V_s = \{1,4\}$