Chapter 9-Graphs

Sunday, January 1, 2023
7:34 PM

Graphs:

A ***Graph*** is denoted as (V,E), where V and E are sets representing the nodes and edges, respectively.
Two nodes connected by an edge are called ***neighbors*** or ***adjacent***.

A graph edge can be ***undirected*** (can be traversed in both directions), or ***directed*** (like relations graphs).

Two nodes can be connected by ***multiple edges***.
A node can also form a ***loop edge*** (starting and ending at itself)

A ***Simple Graph*** has *neither* multiple edges or loop edges.

Degree of a node is the number of edges which a node is an endpoint for.

Handshaking Theorem:
The sum of the degrees of all the nodes is **twice** the number of edges.

$$\sum_{v \in V} deg(v) = 2|E|$$

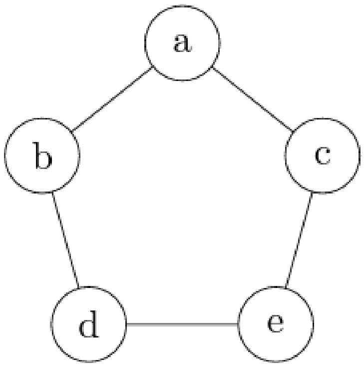
or

$$\sum_{k=1}^{n} v_k \in V deg(v) = 2|E|$$

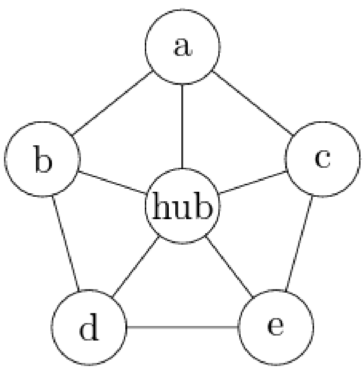
A ***Complete Graph*** on n nodes (shorthand name Kn), is a graph with n nodes in which every node is connected to every other node.

$\sum_{k=1}^{n}(n-k) = \sum_{k=0}^{n-1} k = \frac{n(n-1)}{2}$ edges

Cycle Graph is a graph with edges connecting the nodes one after another and then the last node is connected to the first node. (a cycle graph contains 2n cycles, with n being the number of nodes)



Wheel: A wonkier version of a cycle graph.
The additional central "hub" node that is connected to all other nodes. So the n for a wheel has a slightly different meaning. W_n has **n+1** nodes, and **2n** edges.



Isomorphism:
Given 2 graphs, G1 = (V1, E1) and G2 = (V2, E2),
An ***Isomorphism*** from G1 to G2 is a bijection
$f : V_1 \to V_2$
Such that a function relates all the nodes in V1 and all the nodes in V2, and two nodes in V1 *(say, nodes x and y)* are joined by an edge iff two corresponding nodes *(say, nodes f(x) and f(y))* are joined by an edge.

Graph Isomorphism is like equivalence relation, although looks different, are pretty much the same thing. (same abstract graph)

Properties of Isomorphism:

The two graphs must have the same number of nodes and the same number of edges.

For any node degree k, the two graphs must have the same number of nodes of degree k. For example, they must have the same number of nodes with degree 3.

Proving Two Graphs are NOT Isomorphic can be done by using the above properties (when they are false then it is not isomorphic)

Subgraphs:
If G and G' are graphs, then G' is a subgraph of G if and only if the nodes of G' are a subset of the nodes of G and the edges of G' are a subset of the edges of G.
If two graphs G and F are isomorphic, then any subgraph of G must have a matching subgraph somewhere in F.
To prove that two graphs are NOT Isomorphic, try to find sub graphs in both that are not matching.

Walk:
Going from a specific node to another node, the finite sequence of nodes (or edges).
Closed Walk is the walk with the same starting and ending node.
Open Walk is the opposite.
A ***Path*** is a walk in which **no** node is used **more than once**.
A ***Cycle*** is a closed walk with at least three nodes in which no node is used more than once except the starting and ending node.
An ***Acyclic*** graph does NOT contain any cycles.

A ***Connected Graph*** has a walk between every pair of nodes.
If a graph is not connected, you might be able to divide it into ***Connected Components***, with each component containing the max amount of nodes.

A ***Cut Edge*** is a singular edge that would make a graph no longer connected **if removed**.

Distance: distance, yea…

Diameter of a graph: the maximum distance between any pair of nodes in the graph.

Euler Circuit of a graph is a closed walk that uses **each edge** of the graph exactly once. (Every single edge needs to be used exactly once)
*An Euler circuit is possible exactly when the graph is **connected** and each node has an* **even** ***degree***.

A graph G = (V,E) is a ***Bipartite Graph*** if we can split V into two non-overlapping subsets V1 and V2 such that every edge in G connects an element of V1 with an element of V2.
(we can divide the set of nodes in a way that no edge connects two nodes from the same part of the division)
For example, here's a Bipartite Graph split into groups R and G: