Theorem 1. Every walk contains a path (for graphs or digraphs).

Theorem 2. Every odd closed walk contains an odd cycle (for graphs or digraphs).

Theorem 3. A graph is bipartite if and only if it has no odd cycles.

Theorem 4. A graph is Eulerian if and only if it has at most one nontrivial component and every vertex degree is even.

A digraph is Eulerian if and only if it its underlying graph has at most one nontrivial component and every vertex has indegree equal to its outdegree.

Theorem 5. An edge is a cut-edge if and only if it is contained in no cycles.

Theorem 6. If G is simple and $\delta(G) \geq k$, then G contains a path of length at least k. If $k \geq 2$, then also G contains a cycle of length at least $k+1$.

Theorem 7. For any graph G, $\sum_v d(v) = 2e(G)$. For any digraph D, $\sum_v d^+(v) = \sum_v d^-(v) = e(D)$.

Theorem 8. Every loopless G has a bipartite subgraph with at least $e(G)/2$ edges.

Theorem 9. The maximum size of a triangle-free simple graph on n vertices is $\lfloor n^2/4 \rfloor$.

Theorem 10. A sequence $d_1 \geq \cdots \geq d_n \geq 0$ of integers is the degree sequence of

- some graph if and only if the sum $\sum d_i$ is even.
- some loopless graph if and only if the sum is even and $d_1 \leq \frac{1}{2} \sum d_i$.
- some simple graph if and only if the "residual sequence" from Havel-Hakimi is graphic.

Theorem 11. Any two of (1) "$e(G) = n(G) - 1$", (2) "G is acyclic", (3) "G is connected" imply the third. Also equivalent: that G is loopless and has the property that every pair of vertices is joined by a unique path.

Theorem 12. Every nontrivial tree has at least two leaves. Deleting a leaf from a tree yields a tree. Every edge of a tree is a cut-edge. Adding any missing edge to a tree creates exactly one cycle. Every connected graph has a spanning tree. The center of a tree is either K_1 or K_2.

Theorem 13. Prüfer codes are in one-to-one correspondence with the set of trees with vertex set $[n]$. Hence the number of such is n^{n-2}.

Theorem 14. $\tau(G) = \tau(G - e) + \tau(G \cdot e)$.

Theorem 15. $\tau(G) = (-1)^{s+t} \det Q^*$, where $Q_{i,j}$ is $-a_{i,j}$ if $i \neq j$ and $d(v_i)$ if $i = j$, and Q^* is obtained from Q by deleting row s and column t.

Theorem 16. Kruskal's and Prim's Algorithms solve the Minimum Spanning Tree problem.

Dijkstra's Algorithm solves the Minimum Distances from u problem.