Other Study Materials for Graph Theory:

1. **https://www.brainscape.com/p/2T8KF-LH-8JMSW**
2. **https://docs.google.com/document/u/4/d/1M6U9RPn0JOqlk6FfMdKNzkE1Dyb-gUcaoBcrkqdlTqc/mobilebasic**

Week 11

Topics Covered: **Graphs, Handshake Lemma, Special Graphs, Walks and Paths, Connected Components, Shortest Paths in a Grid**

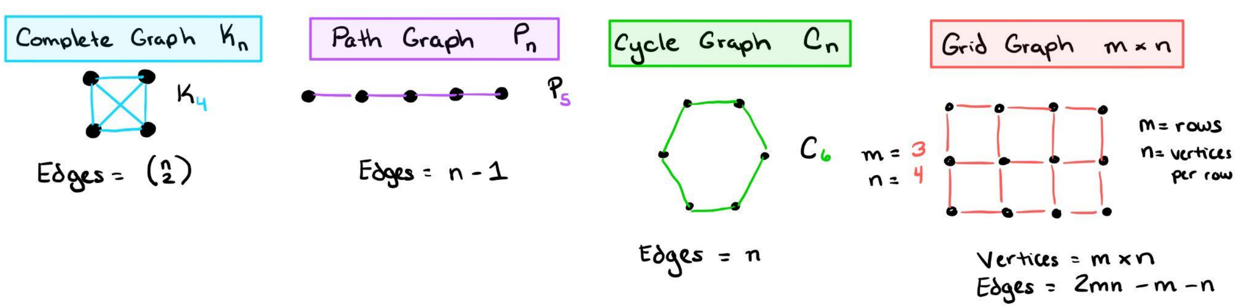
- **Definitions:**
  - **An undirected graph** is a pair G = (V, E) where V is a finite non-empty set of vertices or nodes and E ⊆ 2^{V} is a finite (possibly empty) set of edges consisting only of subsets of cardinality 2.
    - Further clarification:
      - If V = {1, 2, 3}, 2^{V} = {empty set, 1-2, 2-3, 1-3}. These are the sets of vertices in a graph (with cardinality of 2). 2^{V} refers to the powerset of vertices (all such subsets), but we're only interested in subsets with cardinality 2 as an edge can exist between 2 vertices.
      - E ⊆ 2^{V} means edges in G is a subset of all possible connections (and emptyset, which represents no edge). Another way to look at it is, that the set of edges is a subset of pairs of vertices.
      - For example, you can have 3 vertices and no edge at all, or 3 vertices and each is connected to the other two, or 3 vertices and the only edge is 1-2.

        - **Note:** Our definition of an edge as a set of nodes of cardinality 2 precludes“loops” or “parallel edges”.
        - **Not** allowed: 
    - **Edge -** An edge u-v is **incident** to either of its **endpoints** u and v. Two vertices such that u-v are called adjacent (or neighbors).
    - **Degree -** The degree (stylized: deg(u)) of a vertex is the number of neighbors (the number of other vertices adjacent to that vertex). A vertex with degree 0 is called isolated.
    - **Edgeless Graph -** A graph with vertices and no edges.
    - **Complete Graph -** A graph with edges between any two vertices. Notation is K_n where n >= 1. A complete graph has **$n \choose 2$ edges** (the number of unordered pairs), or **n(n-1)/2**.
    - **Path Graph -** A path graph has **n vertices** and **n-1 edges** arranged in a row. The notation is P_n where n >= 1.
        - Note: For n >= 3 we have **two vertices of degree 1 in P_n** and **the rest have degree 2**.
    - **Cycle Graph -** A cycle graph has **n vertices** and **n edges** arranged in a circle. The notation is C_n where n >= 3. **All vertices in a cycle graph have degree 2**. C_1 and C_2 are undefined.
    - **Grid Graph -** An m x n grid graph has m rows of n vertices where each vertex is linked by an edge to the vertices closest to it. **A grid graph has mn vertices**. When m, n >= 3, we have 4 vertices with degree 2 and the rest with degree 3 or 4. **Total edges is 2mn - m - n**.
        - When m,n ≥ 3, we have 4 vertices with degree 2 and the other vertices have degree 3 or 4
        - in an m×n grid graph every path of minimum length from the “lower left corner” to the “upper right corner” has length m+n–2 and traverses edges only “upwards” or “rightwards”
    - **Walk -** A walk is a non-empty sequence of vertices linked by edges. The length of a walk is the number of edges. An isolated vertex is a walk of length 0.
    - **Path -** A path is a walk in which all vertices are distinct. Walks of length 0 are paths.
    - **Well-Ordering Principle -** Every non-empty set of natural numbers has a least element.

  - **Connectivity -** Two vertices are connected if there exists some walk between them. $u- ... -v$ is the connectivity relation.
  - **Connected Component -** A connected component of a graph $G = (V, E)$ is a set of vertices $C \subseteq V$ such that any two vertices in C are connected and there is no strictly bigger set of vertices $C \subsetneq C' \subseteq C$ such that any two vertices in C' are connected (maximally connected).
  - **Reflexive -** For any u we have $u - \cdots - u$ so connectivity is reflexive
    - U is connected to u by the walk/path of length 0
  - **Symmetric -** For any u and v, if $u - \cdots - v$ then $v - \cdots - u$ so connectivity is symmetric.
    - If $u - w1 - \cdots - wn - v$ is a walk, then its reversal $v - wn - \cdots - w1 - u$ is also a walk.
  - **Transitive -** For any u,v, and w, if $u - \cdots - v$ and $v - \cdots - w$ then $u - \cdots - w$ so connectivity is transitive.
    - If $u - y1 - \cdots - ym - v$ and $v - z1 - \cdots - zn - w$ are walks, then their concatenation $u - y1 - \cdots - ym - v - z1 - \cdots - zn - w$ is also a walk.
    - Note that the reversal of a path is also a path, but the concatenation of two paths need not be a path.
  - **Connected Graphs -** A connected graph has one connected component while a disconnected graph has 2 or more. Every connected graph with n vertices has n-1 or more edges.
    - In edgeless graphs, each vertex forms a separate connected component. Edgeless graphs with two or more vertices are disconnected, while 1-vertex graphs are connected.
  - **Partition -** Any two distinct connected components are disjoint. Connected components are a partition of vertices.
    - Suppose, toward a contradiction that two distinct connected components, C1 and C2, are not disjoint, then $w \in C1 \cap C2$. A vertex $u \in C1$ is connected to w. Similarly, a vertex $v \in C2$ is connected to w. By symmetry and transitivity $u - \cdots - v$. But C1 and C2 are maximally connected. Contradiction.
  - **Distance** - The length of the shortest path between two vertices

- **Propositions:**
  - **1: Handshaking Lemma -** The sum of the degrees of all nodes in a graph is twice the number of edges
    - $\sum_{v \in V} \deg(v) \;=\; 2\,|E|$

        - **Proof:** Since each edge is incident to exactly two vertices, each edge contributes two to the sum of the degrees of vertices.
    - **2: Number of Vertices of Odd Degree -** In any graph there are an even number of vertices of odd degree.
        - Since each vertex in Vo has odd degree, for the sum of the degrees of vertices in Vo to be even, |Vo| (the cardinality) must be even.
    - **Walks Contain Paths -** If we have a walk of length $n \geq 3$ then there exists a path of length n at most. When there is a walk, there is a path that is not longer.
        - If u_0−u_1−⋯−u_n−1−un is a walk of length n≥3 such that u_0 != u_n, then there exist vertices v_1,...,v_m such that u_0−v_1−⋯−v_m−u_n is a path whose sequence of nodes and edges is a subsequence of the sequence of nodes and edges of u_0−u_1−⋯−u_{n−1}−u_n. Here, “subsequence” preserves order, but it does not necessarily consist of consecutive elements, i.e., e1e3e4 is a subsequence of e1e2e3e4.
    - Any two distinct connected components are **disjoint**.
    - **In any graph G = (V,E) we have |E| ≥ |V| − |CC|.** Another variation: **|CC| ≥ |V| − |E|**
    - **|E| ≤(|V| choose 2)**
    - The maximum number of edges is attained for the complete graph.
        - Corollary: Every connected graph with n vertices has n−1 or more edges
        - In a connected graph |CC|= 1 therefore |E| ≥ |V| − |CC| = n−1



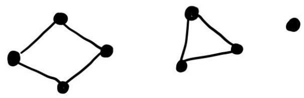
Counting Connected Components

$|CC| \geq |V| - |E|$

number of connected components

number of vertices

number of edges



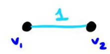
$3 \geq 8 - 7$

Handshaking Lemma

*The sum of the degrees for each vertex is equal to two times the number of edges

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

degrees are the number of edges connected to a vertex



$\deg(v_1) = 1$

$\deg(v_2) = 1$

$2|E| = 2 \cdot 1 = 2$

Two times the number of edges

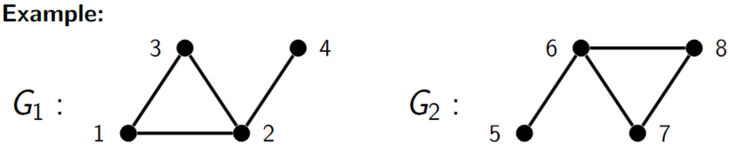
$\boxed{1 + 1 = 2}$ ✓

Week 12

Topics Covered: Subgraphs, Counting Paths, Cycles, Forests, Trees, Leaves, Cut Edges

- **Definitions:**
  - **Isomorphic** - Two graphs G1= (V1,E1) and G2 = (V2,E2) are isomorphic, when there is a bijection β:V1→V2 such that for any u1,v1 ∈ V1 we have u1−v1 ∈ E1 iff β(u1)−β(v1) ∈ E2.

    Example:

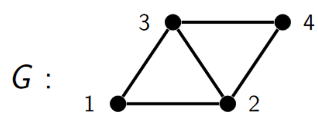
    

    $G_1 \simeq G_2$ by the bijection $4 \mapsto 5,\ 2 \mapsto 6,\ 1 \mapsto 7,\ 3 \mapsto 8$.

    $1 \mapsto 8,\ 3 \mapsto 7$ also works! Note that the bijection must preserve node degree.
  -
  - **Subgraph** - A graph G1= (V1,E1) is a subgraph of the graph G2 = (V2,E2) when V1 ⊆ V2 and E1 ⊆ E2. (Beware: not all pairs of such subsets form graphs!)

- How many paths of length 2 are there in Cn (n≥3)?
    - By the bijection rule there are as many path subgraphs of length 2 as there are vertices. This gives us the answer: n.
- **Induced Subgraph** - If G = (V,E) is a graph and V’⊆ V is a set consisting of some of G’s nodes, the subgraph of G induced by V'is the graph G'= (V',E') where E'consists of all the edges of G whose endpoints are both in V'.
- **Closed Walk** - A walk in which the first and last vertex are the same.
- **Cycle -** A closed walk of at least length of 3 in which all nodes are pairwise distinct, except for the first and last.
    - The length of the cycle is the length of the closed walk.

      Examples of closed walks (cw's) and cycles:

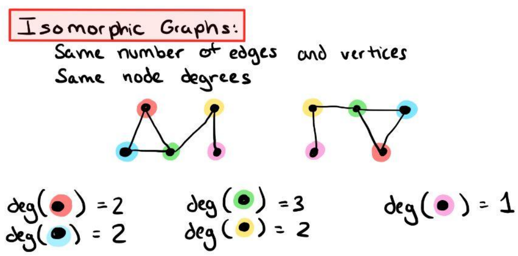
      

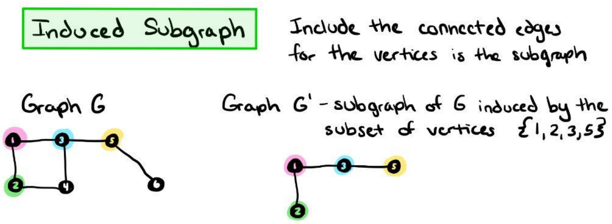
      . 1–2–3–4–2–1 cw, **not** cycle
      . 2–3–4–2 cycle, length 3
      . 2 cw, **not** cycle
      . 2–4–2 cw, **not** cycle
      . 1–2–4–3–1 cycle, length 4
    -
    - How many cycles are there in K4? How many cycles of length 4 in K5? How many total cycles in K5?
        - Total # of cycles in K4:
            - The number of cycles of length 3 is (4 choose 3) = 4 (4 vertices, choose 3 vertices).
            - The number of cycles of length 4 is (3 choose 2) = 3 (3 edges incident to one node, choose 2 of them).
            - Total = 7.
        - Cycles of length 4 in K5:
            - Choose 4 out of 5 vertices = 5 ways.
            - Construct a cycle of length 4 on the 4 vertices. (3 choose 2) = 3.
            - 5x3 = 15 cycles.
        - Total # of cycles in K5:
            - Cycles of length 5: we need a path of length 3 that doesn’t go through the node we picked in 4 choose 2 = 2 ways x (4 choose 2) - choose 2 out of the 4 edges incident to a node. 2x6 = 12.
            - Cycles of length 4: (3 choose 2) = 3 (2 edges incident to one vertex) * (5 choose 4) = 5; choose 4 vertices out of 5. 3x5 = 15.
            - cycles of length 3: (5 choose 3) = 10.

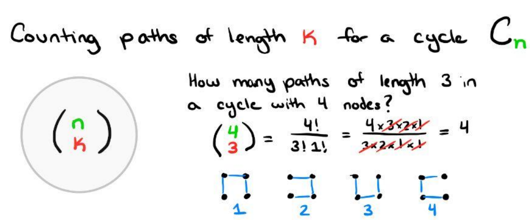
        - 12 + 15 + 10 = 37 cycles.
    - **Connected components** - Let G= (V,E) be a graph. We defined connected components(cc's) as subsets $C \subseteq V$ and observed that they form a partition of V. Now, we regard a cc C as a graph, namely the subgraph of G induced by the set of vertices in C. The cc's also partitions the set of edges E.
    - **Acyclic** - A graph with no cycles. The connected components of an acyclic graph are also acyclic.
        - An acyclic graph is called a **forest** since all its cc's are trees.
    - **Tree** - A graph that is both connected and acyclic.

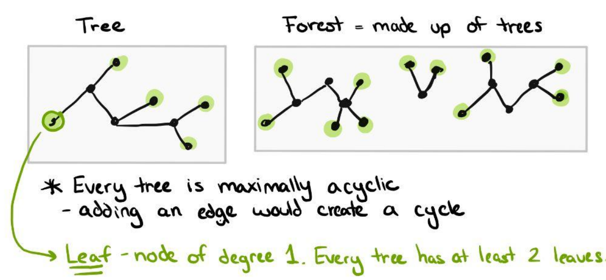
- **Propositions:**
    - Any two complete graphs, two path graphs, two cycles graphs, two edgeless graphs, or two mxn grids are isomorphic **if and only if** they have the **same number of vertices**. An m×n grid is also isomorphic to any nxm grids.
        - A path graph on n vertices is a graph isomorphic to Pn. Hence, a path graph of length l is a graph isomorphic to Pl+1.
    - Every edge ${u, v} \in E$ belongs to exactly one of the subgraphs induced by the connected components of G
    - To count the number of paths of length k in the graph G, add up the number of paths of length k in each of its connected components
    - Let G1 and G2 be two isomorphic graphs, then:
        - G1 is acyclic if and only if G2 is acyclic
        - G1 is connected if and only if G2 is connected
        - G1 is a tree if and only if G2 is a tree
    - A tree has one more vertex than edge, |E| = |V| - 1
    - If G = (V, E) is a forest, then |E| = |V| - |CC|
    - Every tree with edges has at least one leaf (actually, at least two!).
    - Every tree is minimally connected (removing an edge disconnects a tree)
    - Every tree is maximally acyclic (adding an edge between two non-adjacent nodes creates a cycle)
    - **Any two distinct vertices of a tree are connected by a unique path**
    - Every tree is unique-path connected (any two distinct vertices of a tree are connected by a unique path)
    - Adding an edge to an acyclic graph creates at most one cycle
        - If it is a forest with at least two trees, the added edges can go between nodes in these trees.

- Any closed walk of non-zero length that traverses at least one of its edges exactly once contains a cycle
- A graph such that any two distinct vertices are connected by a unique path must be a tree
- Removing a cut edge increases the number of connected components by exactly 1









Week 13

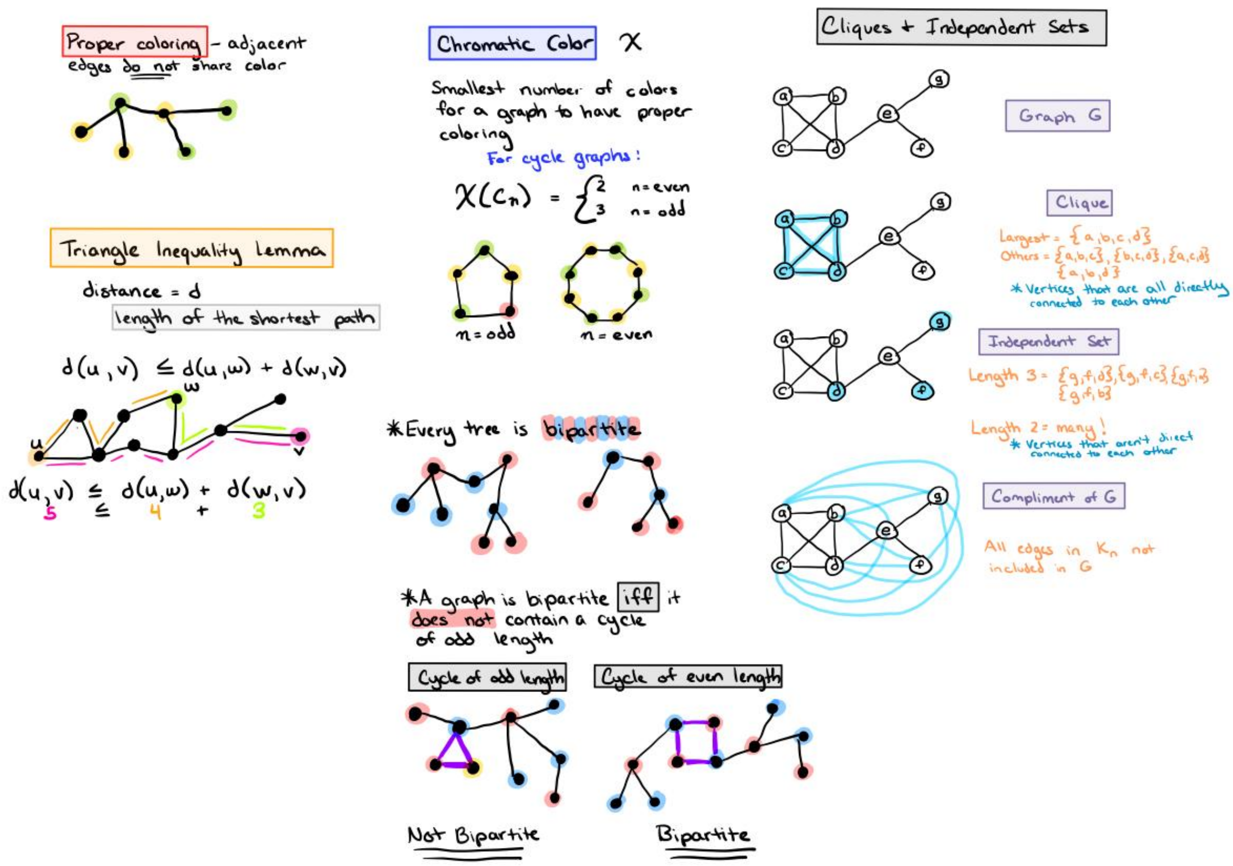
Topics Covered: **Spanning Trees, Graph Coloring, Colorability and maximum degree, No odd cycles, Bipartite Graphs, Cliques,   Sets**

- **Definitions:**
  - **Spanning Subgraph** - Of the graph G = (V, E) is a subgraph whose vertex set is the entire set V.
  - **Spanning Tree** - of a connected graph G is a spanning subgraph that is a tree.

- **Spanning Forest** - of a graph G consists of a spanning tree for each of the connected components of G.
- **K-Coloring** - of a graph G = (V,E) with is a function $f : V \to [1..k]$.
- **Proper Coloring** - for any edge u–v we have f(u) != f(v)
- **K-Colorable** - A graph that admits a proper k-coloring
    - Note that k-colorable implies j-colorable for any $j \geq k$.Clearly, a graph with n vertices is n-colorable.
- **Chromatic Number** - The smallest k such that G is k-colorable. It is denoted by $\chi(G)$.
- **Bipartite Graph** - 2-colorable graphs are also this.
- The **distance** between two vertices $u,v \in V$, notation d(u,v), is the length of a shortest path from u to v.
- **Complete Subgraph** - A subgraph that is isomorphic to the complete graph Kn for some n.
    - **Clique** - is used both for such a complete subgraph of G and for a subset gof the vertices of G that induces a subgraph that is complete.
        - Alternative definition: A subset of vertices any two of which are adjacent.
        - **Size of a clique** - Its number of vertices
- **Independent Set** - A subset of vertices $S \subseteq V$ when no two vertices in S are adjacent.
    - Alternative definition: The induced subgraph is edgeless.
- **Complement** - of G is the graph G = (V, E) where E_bar = {{u,v} | u,v ∈ V ∧ u != v ∧ {u,v} \not-in E}

- **Propositions:**
    - Every connected graph has a spanning tree.
    - Every graph has a spanning forest.
    - Spanning trees always exist.
    - Removing a **cut edge** in a graph increases the number of connected components by EXACTLY one.
    - An edge is a cut edge **if and only if** it does not belong to any cycle.
    - A connected graph is a tree **if and only if** each one of its edges is a cut edge.
    - $\chi(G) = 1$ iff G is edgeless.
    - For $n \geq 2$ we have $\chi(Pn) = 2$.
    - $\chi(Cn) = 2$ when n is even; = 3 when n is odd
    - $\chi(Kn) = n$.

        - For n ≥ 2, any k-coloring with k<n cannot be proper. Indeed, by the Pigeonhole Principle, at least two vertices get the same color. But in Kn any two vertices are adjacent.
    - All path graphs are bipartite and a cycle graph is bipartite if and only if it has an even number of nodes.
    - Every tree is bipartite.
    - Every graph is $\Delta(G) + 1$-colorable where $\Delta(G)$ is the maximum degree of a node in G.
    - A graph is bipartite if and only if it does not contain a cycle of odd length.
    - Every subgraph of a bipartite graph is also bipartite.
    - If S is a subgraph of G, then $\chi(S) =< \chi(G)$
    - **Triangle inequality: d(u, v) ≤ d(u, w) + d(w, v)**
        - Concatenating these two paths gives us a walk of length d(u,w) + d(w,v). But d(u,v) is the length of a shortest path from u to v and hence it is ≤ than the length of any walk from u to v.
    - A graph has a proper coloring iff each of its connected components have a proper coloring.
    - A graph is k-colorable if and only if its set of vertices can be partitioned into k independent sets.
    - The leaves of a tree form an independent set
    - Let G = (V, E) and $S \subseteq V$ be a set of vertices, S is a clique in G if and only if it is an independent set of G.
        - For any u,v ∈ S, we have u−v inG iff¬(u−v) in G_bar.
        - Concepts of clique and of independent set are dual.
    - How many edges does the complement of Pn have?
        - We have that Kn has n(n−1)/2 edges, and Pn has n−1 edges. Thus the complement of Pn has n(n−1)/2 − (n−1) = n(n−1)−2(n−1)/2 = (n−2)(n−1)/2.



Week 14

Topics Covered: Directed graphs, Reachability, Strong Connectivity, Directed Acyclic Graphs (DAGs), Topological sorting, Binary trees

- **Definitions:**
  - **Directed graph (digraph) -** G = (V,E) consists of a non-empty set V and a set E is the subset of V x V of edges which are ordered pairs of vertices.
    - This allows **self-loops**(v,v) ∈ E where v ∈ V but not for parallel edges.We can have **anti-parallel edges** (u,v),(v,u) ∈ E where u,v ∈ V.

        - We use the notation u→v for both the edge (u,v) itself and for the fact that (u,v) ∈ E. u→v is an **outgoing** edge from u and an **incoming** edge to v.
        - V is a **successor** of u and u is a **predecessor** of v. u and v are neighbors, when u→v or v→u, a symmetric relationship.
        - Since we allow self-loops, a vertex can be its own neighbor in which case it is also its own predecessor and its own successor.
    - **Isolated Vertices** - vertices with no neighbors.
    - **Edgeless digraph** - just like an edgeless graph.
    - **Outdegree** of a vertex u is the number of successors of u and the number of outgoing edges from u : out(u) (A node of outdegree 0 = **Sink**)
    - **Indegree** of a vertex u is the number of predecessors of u and the number of incoming edges to u : in(u) (A node of indegree 0 = **Source**)
    - **deg(u) = out(u) + in(u)**
    - **Directed Walk -** A non-empty sequence u0,u1,...,uk such that u0→u1→⋯→uk. We call this a directed walk from u0 to uk of length k.
        - Do not confuse the directed walk of length 0,u with the directed walk of length 1 given by the existence of a self-loop: v→v. Walks of length 0 arepaths but walks of length 1 given by self-loops are not paths.
    - **Directed Path** - A walk with no repeated vertices.
        - For every vertex v there is a directed path of length 0:v.
    - **The length of the directed cycle = k+1 (u0 -> u1 -> u2 -> ... -> uk -> u0)**
    - **Directed Cycle** - A closed walk u0→⋯→uk→ u0, with u0,...,uk all distinct. The length of the cycle is k + 1.
        - There are no cycles of length 0. A self-loop gives a cycle of length 1. A cycle of length 2 consists of two vertices and edges between them in opposite directions (antiparallel edges).
    - **Reachability** - A vertex v is **reachable** from a vertex u when there is a walk (and therefore a path) from u to v. We write u→→ v for the **reachability relation**.
        - The reachability relation is reflexive, i.e.,u→→u and transitive, i.e., u→→v and v→→w imply u→→w.
        - For reflexivity consider walks of length 0. For transitivity we concatenate walks.The relation u→→v is called the **reflexive-transitive closure** of the (edge) relation u→v.
    - **Strongly Connected Components** - The maximally strongly connected sets of vertices

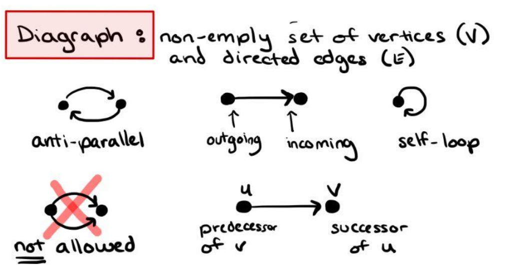
- **Reduced Graph** - Given a digraph G= (V,E) its reduced graph has as vertices the scc's of G and as edges the pairs (S1,S2) where S1 and S2 are distinct scc's such that there exist u1 ∈ S1 and u2 ∈ S2 such that u1→u2 is an edge in G.
- **Directed Acyclic Graph** - a digraph without directed cycles (not even of length 1).
- **Topological Sort** - of a digraph is a sequence σ in which every vertex appears exactly once (i.e., a permutation of its vertices) such that for any edge u→v in the graph, the vertex u appears in σ before (but not necessarily immediately before) the vertex v.
- **"Fluid" DAG** - If it has no isolated vertices, and if there is a directed path from every source to every sink.
  - The minimum number of edges in a fluid DAG with 2 sources and 3 sinks is 5:
    - In addition to the 2 sources and 3 sinks we have an intermediate node through which all the source-to-sink paths go.
- **Rooted Tree** - a pair (T,r) where T= (V,E) is a tree and the vertex r ∈ V is designated as a root.
  - **Binary Rooted Tree** - Every node has at most two children
  - The single node of a one-node rooted tree is a root. It is also a leaf, although it has degree 0. The one-node rooted tree is a complete binary tree.
- **Child** - Successor
- **Parent** - Predecessor
- **Height** - Distance from root to farthest leaf
- Binary Search Tree - Are ordered. They may have a left child and a right child. The left child has a value less than the parent. The right child has a value more than the parent.
- **Complete vs. Full Binary Tree** - Full: Every non-leaf node has exactly 2 children. Complete: It is full and all leaves are the same distance from the root.

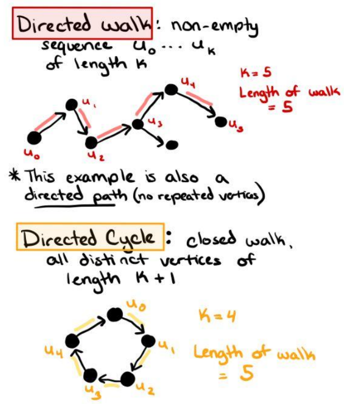
- **Propositions:**
  - Maximum number of edges that a digraph with n nodes can have is n^2. Each node can have n successors, i.e. n edges leaving it to every other node and itself. Since there are n nodes, then the maximum number of edges for a digraph with n nodes is n^2.

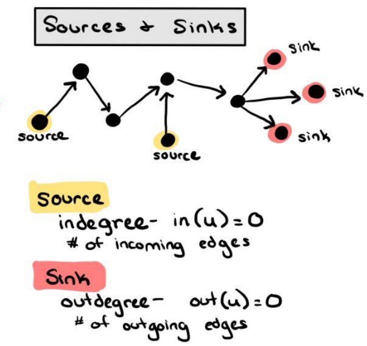
- The **sum of the outdegrees** of all vertices in a directed graph equals the **sum of the indegrees** of all vertices and further equals the **number of edges**
- The reachability relation is **reflexive**, i.e. u ->> u and **transitive**, i.e. u ->> v and v ->> w implies u ->> w.
- Strong connectivity is:
  - **Reflexive**: u <<->> u
  - **Symmetric**: u <<->> v => v <<->> u
  - **Transitive**: u <<->> v and v <<->>w => u <<->>w
- A set of vertices is strongly connected when any two of its vertices are strongly connected.
- Any two distinct strongly connected components are **disjoint.**
- The reduced graph has **no directed cycles.**
- Because the reduced graph has edges only between distinct vertices we cannot have cycles of length 1.
- If a digraph has a topological sort then:
  - The first vertex in the sort is a source and the last vertex is a sink
  - The digraph is a DAG
- Every DAG has **at least one** source and **at least one** sink
  - Isolated vertices are both sources and sinks
- Every DAG has **at least one** topological sort
- How many distinct topological sorts does such an edgeless digraph have?
  - Since there are no edges in the graph, there are no constraints in the ordering of the nodes in the topological sorts; therefore the number of topological sorts is equal to the number of ways to order n nodes, i.e.n!.
- Any edge of a rooted tree is traversed in **the same direction** by all unique paths from the root to each of the other vertices
- A binary tree of height h has a maximum of $2^{h+1} - 1$ nodes among which are 2^{h} leaves. This maximum is attained for the complete binary tree of height h.
  - We define a **complete binary tree of height h** to be a rooted tree in which every non-leaf node has two children and all leaves are at distance h from the root.
- How many edges are in a complete binary tree of height h?
  - We compute the sum of indegrees: We have that every node except the root has exactly one incoming edge. It follows that the number of edges is (2^{h+1}–1) −1 = 2^{h+1} – 2.
  - We reach the same answer by computing the sum of outdegrees instead: we have that every node except the leaves have two outgoing edges. It

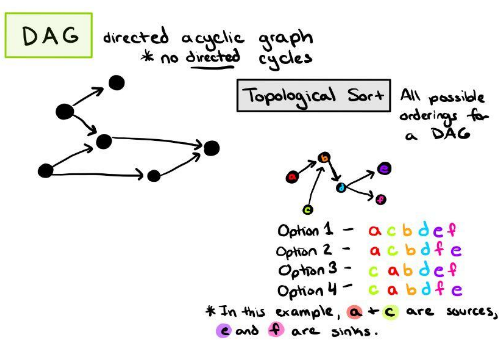
follows that the number of edges is 2(2^{h+1} – 1 – 2^{h}) = 2(2^{h–1}) = 2^{h+1} – 2.

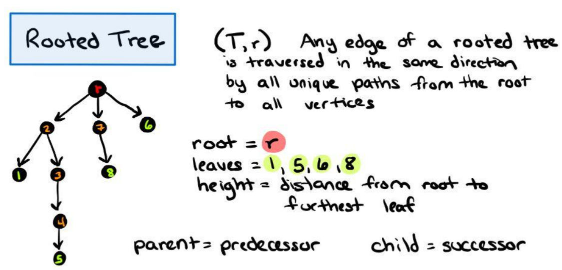
- **Corollary:**
  - The strongly connected components determine a partition of the vertices **(but not of the edges)**



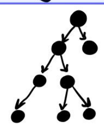






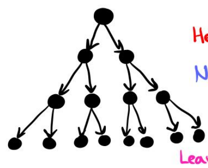


Binary Rooted Tree - every node has at most two children



* A binary tree of height h has a maximum $2^{h+1}-1$ nodes among which are 2^h leaves

Complete Binary Tree



Height = 3

Nodes = $2^{3+1}-1$
$= 2^4 - 1$
$= 16 - 1$
$= 15$

Leaves = 2^3
$= 8$

Solving Proof Questions:

- Always try direct proof first (unless it’s obvious you can’t).
- If direct proof doesn’t work, try contradiction (or contrapositive).
  - For if and only if statements, you cannot use induction. Use proof by contradiction and/or contrapositive.
- If contradiction/contrapositive don’t work, move to proof by induction.
  - If you have a “there exists” question, you can go from k to k+1 and complete the proof.
  - If you have a “for any” or “for all”, you start at k+1 and remove a node or edge (depending on what the question is asking) and then use your induction hypothesis for k. Then add the removed node (or edge) back.