Tree (Connected Acyclic Graph)

Ex:

Why study trees?

(1) Algorithmic implications → a subset of a tree is a tree hence trees are easy

→ a subset of a graph is not a tree

Imp: Graph theory of graph independent of drawing

(2) Data structure

Tree = more generalized version of L.L

Lemma: Let $T = (V, E)$, $|V| \geq 1$. Then T has at least 2 leaves.

Proof: Then there is longest path P of T

$v_1 v_2 v_3 \dots v_{k-1} v_k$

If v_1 v_k are not leaves, then there exists a longer path, which contradicts our assumption.

Hence v_1 and v_k have to be leaves.

Lemma 2: The following statements are equivalent for a graph G of n vertices (or equivalent)

(1) G is a tree.

(2) G is a cycle free — removing an edge e makes it not connected, $\Rightarrow$ G is disconnecting tree

Adding a new edge of edge e the graph will be connected and acyclic.

Lemma 3: Connected, acyclic, n edges, n vertices

Proof: Let T be a tree of G (spanning tree)

Theorem 1: The following statements are equivalent for a graph $G = (V, E)$

(1) G is a tree

(2) G is cycle free

(3) For every vertex $v \in V$ there exists an unique path from v to v

(4) G is connected & deleting an edge makes it disconnected

(5) G has no cycle & $v \in E$ adds a cycle

Hence $G \in \binom{V}{2}$ $\leftarrow$

Proof: $1 \to 2$

If G is a tree then it's acyclic, which assumes unique path exists between any v, w in V.

$(2) \to (3)$

G is connected

If G is not $\Rightarrow$ there $\Rightarrow$ G has a cycle

any two points on the cycle doesn't have a unique path.

(contradiction)

$(3) \to (4)$

G is connected

If G is not $\Rightarrow$ there $\Rightarrow$ G has a cycle

Hence there exists a path between every v, w

$\Rightarrow$ G is connected

$(4) \to (5)$

G is a tree

Hence true. Hence for $n \geq 2$

Induction hypothesis - any graph with less than $|V|$ vertices satisfies (5).

Induction step - Consider $G = v$, where $v \geq 2$ and then it satisfies induction hypothesis

$\Rightarrow |V| = 1 \Rightarrow |E(G-v)| = 0$

$\Rightarrow |E| = 1 = n$

$\Rightarrow \boxed{|E| = |V|}$

Proof: $(5) \to (1)$

$|E| = |V|$

Sum of degrees of $G = 2|E| = 2n$

So vertex of deg 1

Ex: If $\deg(v) \geq 2$, $w \in V$

$\Rightarrow \sum_{v \in V} \deg(v) \geq 2n$

which contradicts our $\sum \deg(v) = 2n$

$\therefore$ deg $(v) = 1$ and $v \in V$

Now apply induction, since we have a leaf $(v) \to (1)$

Tree not a long graph ; reverse cycle free

↳ cycle, always bad graph

• Edge also subtract leaf.

Spanning Tree of a graph $G = (V, E)$

A subgraph $T = (V, E)$ which is a tree.

Algo 1 (Spanning Tree),

Given $G = (V, E)$

Initialize $E_T = \emptyset$

while T has edges do —

$E_T = \begin{cases} E_T \cup \{e\}, & \text{if graph } (V, E_T \cup \{e\}) \text{ has no cycle} \\ E_T, & \text{o/w} \end{cases}$

Lemma: If has no cycle $\Leftrightarrow |E| = |V| - 1$

$\Leftrightarrow$ G is a tree

G is not connected $\Leftrightarrow$ T has $\geq$ connected components

$|V| - 1$ $\quad |V| - 1$ $\quad \cdots$ $\quad |V| - 1$

$\therefore |V| - k$

Thm (Correctness proof of Algo 1)

If Algo 1 produces an output T with n edges then T is a spanning tree.

If T has n edges then G is disconnected graph with n connected components.

Proof: T has $k < n$ edges $\Rightarrow$ T has $n-k$ connected components.

T is disconnected (≥ 2 components)

$|V| - k \geq 2$

$T = n \quad |V| - n$

$\boxed{n \geq |V| - 2}$

$\Rightarrow G$ has at most $n-1$ connected components.

G has n connected components $\Rightarrow n = 1$

$\boxed{k = n - 1}$, $\boxed{k = n - 1}$

Thm G has vertices x, y that lie in the same component $\Rightarrow$ in T they lies in different components.

If T_1, e was removed by the algo, then

$T \cup e$ are forming a cycle

But was assumption was that x and y lies in the same component in the original graph.

$\therefore$ $T \cup e$ has formed a cycle

$\therefore$ x, y lies in different components initially each other.

$\therefore$ x, y lies in path, then only adding e will make a cycle.

Algo 2: Given $G = (V, E)$ (Spanning a Spanning tree).

Initialize $E_T = \emptyset$, $v_1 \in V$

at the ith iteration :-

if $\exists$ an edge $\{v_i, y\} \in E \setminus E_T$

$\wedge$ $E_T \cup \{y\} \in V_{i+1}$

then

$V_i = V_{i-1} \cup \{y\}$

$E_T = E_{i-1} \cup \{v_i, y\}$

else $i = i + 1$

Thm (Correctness of Algo)

If Algo 2 produces a graph T with n vertices then T is a spanning tree of G, note T is the spanning tree of the component to G containing the initial vertex.

Weighted graphs $G = (V, E, w)$

where $w : E \to \mathbb{R}$ is a weight function & every edge of the graph G

MST (Minimum Spanning Tree)

$G = (V, E, w)$ - connected, find a spanning tree $T = (V, E)$ of the graph G s.t. the sum $w(E') := \sum_{e \in E'} w(e)$ is minimum.

Kruskal's Algo

Let $e_1, e_2, \dots, e_{|E|}$ be the ordering of edges s.t. $w(e_1) \leq w(e_2) \leq \dots \leq w(e_{|E|})$

then run Algo 1.

Prim's Algo :

In Algo 2 while choosing an edge $\{v_i, y\} \in E$ where $v_i \in V_{i-1}$ & $y \in V \setminus V_{i-1}$ choose the edge with minimum w.t.

(Proof of Kruskal's Algo) :-

Lemma: Let (V, E_1) & (V, E_2) be two cycles

$|V| = k$ & $|E_1| > |E_2|$ then $\exists$ $e \in E_1$ which connects two distinct components of the graph (V, E_2)

Proof: Let $V_1, V_2, \dots, V_k$ are components of (V, E_2)

then $|E_2| = |V| - k$

$\left| E_2 \cap \binom{V_i}{2} \right| \leq |V_i| - 1$ $\quad \left[\binom{V_i}{2} \subseteq E(V_i) \right]$

$|V| \to$ edges of (V, E_2) lie in components of (V, E_2)

Let T be the spanning tree of Kruskal's & T' be the spanning tree we work when $w(T) \leq w(T')$

Let the edges of T be denoted by $e_1, e_2, \dots, e_{k-1}$ & T' be $e'_1, e'_2, \dots, e'_{k-1}$

If $w(e_1) \leq w(e_2) \leq \dots \leq w(e_{k-1})$

$w(e'_1) \leq w(e'_2) \leq \dots \leq w(e'_{k-1})$

claim: $w(e_i) \leq w(e'_i)$ $\forall i$

If $\exists$ s.t. $w(e_j) > w(e'_j)$, take the j with minimum value

$w(e_j) > w(e'_j)$, j is the smallest index

$E' = \{e'_1, e'_2, \dots, e'_{j-1}\}$ → subgraph of T'

$E'' = \{e'_1, e'_2, \dots, e'_{j-1}\} \cup \{e_j\}$ → subgraph of T'

$\exists$ $e \in E''$ s.t. the graph $(V, E' \cup \{e\})$ has no cycle

$\boxed{w(e) \leq w(e_j) < w(e'_j)}$

↳ this is contradiction to the def. of e'_j