CS34110 Discrete Mathematics and Graph Theory

UNIT – III, Module – 2

Lecture 34: Graph/tree Connectivity

[Spanning tree; Rooted spanning tree; Weight, rank, nullity, distance; Fundamental circuit; Minimum spanning tree; Construction algorithms]

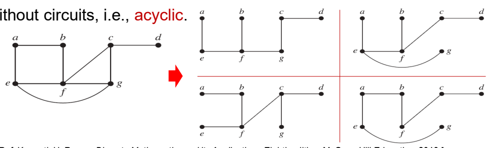
Dr. Suddhasil De

Notation table

Symbol / Notation	Meaning
$d(\mathcal{T}_1, \mathcal{T}_2)$	Distance between spanning trees $\mathcal{T}_1$, $\mathcal{T}_2$ of connected undirected graph $\mathcal{G}$.
$\max d(\mathcal{T}_i, \mathcal{T}_j)$	Maximum distance between any pair of spanning trees $\mathcal{T}_i$, $\mathcal{T}_j$ of connected undirected graph $\mathcal{G}$.
$\mathcal{T}_0$	Central tree of connected undirected graph $\mathcal{G}$.

Graph connectedness

- Spanning tree: undirected connected spanning subgraph without circuits of given graph.
  - Property:: Spanning tree of undirected connected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$: spanning graph $\mathcal{T} = (\mathbf{V}, \mathbf{e})$ of same vertex set, such that $\mathcal{T} \subset \mathcal{G}$, and without circuits, i.e., acyclic.



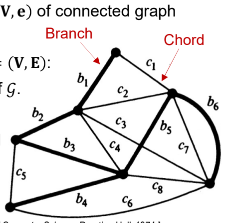
[Ref: Kenneth H. Rosen, *Discrete Mathematics and its Applications*, Eighth edition, McGraw-Hill Education, 2019.]

Graph connectedness

- Spanning tree:
  - Property:: Spanning tree: also called skeleton of given connected graph, scaffolding of given connected graph, maximal tree subgraph of given connected graph, maximal tree of given connected graph.

Graph connectedness

- Spanning tree:
  - Property:: Branch of spanning tree $\mathcal{T} = (\mathbf{V}, \mathbf{e})$ of connected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$: edge in $\mathcal{T}$.
  - Property:: Chord of connected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$: edge of $\mathcal{G}$ not in given spanning tree $\mathcal{T}$ of $\mathcal{G}$.
  - Property:: For two spanning trees $\mathcal{T}_1 = (\mathbf{V}_1, \mathbf{E}_1)$, $\mathcal{T}_2 = (\mathbf{V}_2, \mathbf{E}_2)$ of connected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, possibility of branch of $\mathcal{T}_1$ to become chord with respect to $\mathcal{T}_2$.



[Ref: Narsingh Deo, *Graph Theory with Applications to Engineering and Computer Science*, Prentice-Hall, 1974.]

Graph connectedness

- Spanning tree:
  - Property:: Chord set of spanning tree $\mathcal{T} = (\mathbf{V}, \mathbf{e})$ of undirected connected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$: denoted by $(\mathcal{G} \oplus \mathcal{T})$ or $(\mathcal{G} \setminus \mathcal{T})$, where $(\mathcal{G} \oplus \mathcal{T}) = (\mathbf{V}, \bar{\mathbf{e}})$, $(\mathcal{G} \oplus \mathcal{T}) \subset \mathcal{G}$, such that $\bar{\mathbf{e}} = \mathbf{E} \setminus \mathbf{e} = \{e \in \mathbf{E} \mid e \notin \mathbf{e}\}$, i.e., $\bar{\mathbf{e}}$ = collection of chords.
  - Property:: Chord set: also called tie set of spanning tree, cotree of spanning tree.

Graph connectedness

- Spanning tree:
  - Property:: Rooted spanning tree of digraph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$: rooted tree $\mathcal{T} = (\mathbf{V}, \mathbf{e})$, such that $\mathcal{T} \subset \mathcal{G}$, and every vertex of $\mathcal{G}$ to be endpoint of one of arcs in $\mathcal{T}$.
  - Property:: Spanning forest of (connected/disconnected) undirected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$: forest $\mathcal{F} = (\mathbf{V}, \mathbf{E})$, containing every vertex of $\mathcal{G}$, such that any arbitrary pair of vertices $u, v \in \mathbf{V}$ connected by path P to belong to tree $\mathcal{T} = (\mathbf{V}', \mathbf{E}')$ of $\mathcal{F}$., i.e., $u, v \in \mathbf{V}'$ and $\mathcal{T} \subset \mathcal{F}$.
  - Property:: Disconnected graph with k components to have spanning forest consisting of k spanning trees.

Graph connectedness

- Spanning tree:
  - Property:: Weight of spanning tree $\mathcal{T} = (\mathbf{V}, \mathbf{e})$ of connected undirected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$: $wt(\mathcal{T}) = \sum_{e \in \mathbf{e}} wt(e)$ = total weight obtained by adding weights of edges in $\mathcal{T}$.

Graph connectedness

- Spanning tree:
  - Property:: ⟨Theorem⟩: Simple graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ to be considered as connected graph, if and only if $\mathcal{G}$ to contain at least one spanning tree $\mathcal{T}$.
    [One part of above theorem restated as:: ⟨Theorem⟩: Every connected graph to contain at least one spanning tree.]

Graph connectedness

- Spanning tree:
  - Property:: ⟨Theorem⟩: Simple graph $\mathcal{T} = (\mathbf{V}, \mathbf{E})$ considered to be connected if and only if $\mathcal{T}$ to be spanning tree.

Graph connectedness

- Spanning tree:
  - Property:: ⟨Theorem⟩: For connected undirected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ with $|\mathbf{V}| = n$ vertices and $|\mathbf{E}| = m$ edges, any of its spanning trees to contain $n - 1$ tree branches and $m - n + 1$ chords.

Graph connectedness

- Spanning tree:
  - Property:: For connected undirected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ with $|\mathbf{V}| = n$ vertices and $|\mathbf{E}| = m$ edges, following to be satisfied —

    (i) $rank(\mathcal{G})$ = number of branches in any spanning tree (or forest) of $\mathcal{G}$;

    (ii) $\mu(\mathcal{G})$ = number of chords in $\mathcal{G}$;

    (iii) $rank(\mathcal{G}) + \mu(\mathcal{G})$ = number of edges in $\mathcal{G}$.

Graph connectedness

- Spanning tree:
  - Property:: Distance between spanning trees $\mathcal{T}_1 = (\mathbf{V}_1, \mathbf{E}_1)$, $\mathcal{T}_2 = (\mathbf{V}_2, \mathbf{E}_2)$ of connected undirected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$: $d(\mathcal{T}_1, \mathcal{T}_2) =$ $= {}^1\!/_2 \cdot |\mathbf{E}_1 \,\Delta\, \mathbf{E}_2| = {}^1\!/_2 \cdot |\{e \in \mathbf{E} \mid (e \in \mathbf{E}_1) \oplus (e \in \mathbf{E}_2)\}|$ = half of number of edges not common to $\mathcal{T}_1$ and $\mathcal{T}_2$.
  - Property:: Maximum distance between any pair of spanning trees $\mathcal{T}_i = (\mathbf{V}_i, \mathbf{E}_i)$, $\mathcal{T}_j = (\mathbf{V}_j, \mathbf{E}_j)$ of connected undirected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$: $\underset{\max}{d(\mathcal{T}_i, \mathcal{T}_j)} = \min\{rank(\mathcal{G}), \mu(\mathcal{G})\}$.

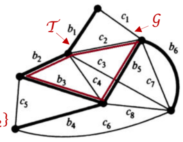
Graph connectedness

- Spanning tree:
  - Property:: ⟨Theorem⟩: Any connected undirected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ to be considered as tree, if and only if adding one edge between any two vertices in $\mathcal{G}$ resulting in formation of exactly one circuit.

Graph connectedness

- Spanning tree:
  - Property:: Fundamental circuit of undirected connected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$: circuit C formed by adding any single chord e of given spanning tree $\mathcal{T} = (\mathbf{V}, \mathbf{e})$ of $\mathcal{G}$, where $e \in \mathbf{E}$, $e \notin \mathbf{e}$, to $\mathcal{T}$.
  - Example-1:: Given graph $\mathcal{G}$: 8 fundamental circuits of $\mathcal{G}$, corresponding to spanning tree $\mathcal{T}$ of $\mathcal{G}$, each circuit formed by one chord (together with branches of $\mathcal{T}$).

Fundamental circuit after adding c_2: $C = \{b_2, b_3, b_5, c_2\}$



[Ref: Narsingh Deo, *Graph Theory with Applications to Engineering and Computer Science*, Prentice-Hall, 1974.]

Graph connectedness

- Spanning tree:
  - Property:: Count of fundamental circuits of given graph $\mathcal{G}$ = invariant of $\mathcal{G}$.
  - Property:: Count of total circuits of given graph $\mathcal{G}$ = invariant of $\mathcal{G}$.
  - Property:: Fundamental circuit defined only with respect to specific spanning tree of graph.
  - Property:: Possibility of given circuit to be fundamental with respect to one specific spanning tree of given graph, but not with respect to different spanning trees of same graph.

Graph connectedness

- Spanning tree:
  - Algorithm:: ⟨Steps to find single Spanning tree of graph⟩:

    Input: Connected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ ($|\mathbf{V}| = n$), $\mathcal{A}dj_{\mathcal{G}} = (a_{ij})_{1\le i,j\le n}$

    Output: Spanning tree $\mathcal{T} = (\mathbf{V}, \mathbf{e})$.

    Step-1: Return $\mathcal{G}$ as $\mathcal{T}$, if no circuit in $\mathcal{G}$.

    Step-2: If at least one circuit present C in $\mathcal{G}$, delete edge e ($e \in C$) from C, thus $\mathcal{G} = \mathcal{G} - e$, and maintaining $\mathcal{G}$ connected.

    Step-3: Repeat step-2, till $\mathcal{G}$ to contain no circuits, and still $\mathcal{G}$ to remain connected.

    Step-4: Return $\mathcal{G}$ as $\mathcal{T}$.

Graph connectedness

- Spanning tree:
  - Property:: Depth-first search algorithm, Breadth-first search algorithm also to be used to find single spanning tree of graph.

Graph connectedness

- Spanning tree:
  - Property:: Elementary tree transformation: procedure for generation of one spanning tree (of connected undirected graph) from another, through addition of one chord and deletion of one appropriate branch, so that no circuit to be found in new spanning tree.
  - Property:: Elementary tree transformation: also called cyclic interchange.
  - Property:: $d(\mathcal{T}_1, \mathcal{T}_2)$ = minimum number of elementary tree transformations involved in transforming spanning tree $\mathcal{T}_1$ to $\mathcal{T}_2$.

Graph connectedness

- Spanning tree:
  - Algorithm:: ⟨Steps to find all Spanning trees of graph⟩:

    Input: Connected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ ($|\mathbf{V}| = n$), $\mathcal{A}dj_{\mathcal{G}} = (a_{ij})_{1\le i,j\le n}$

    Output: All spanning trees $\mathbf{T} = \{\mathcal{T} | \mathcal{T} = (\mathbf{V}, \mathbf{e})$ to be spanning tree$\}$.

    Step-1: Initialize $\mathbf{T} = \emptyset$.

    Step-2: Invoke last algorithm to get $\mathcal{T}$ as spanning tree; $\mathbf{T} = \mathbf{T} \cup \mathcal{T}$.

    Step-3 ⟨Elementary tree transformation⟩: (i) For any tree $\mathcal{T} \in \mathbf{T}$, generate chord set $(\mathcal{G} \oplus \mathcal{T})$ of $\mathcal{T}$; (ii) Generate spanning tree $\mathcal{T}'$ from $\mathcal{T}$ by adding anyone chord c from set $(\mathcal{G} \oplus \mathcal{T})$ to $\mathcal{T}$, and removing one branch b from $\mathcal{T}$, ensuring acyclic.

(contd. to next slide)

Graph connectedness

- Spanning tree:
  - Algorithm:: ⟨Steps to find all Spanning trees of graph⟩: contd.

    Step-4: If $\mathcal{T}' \notin \mathbf{T}$, then $\mathbf{T} = \mathbf{T} \cup \mathcal{T}'$.

    Step-5: Repeat step-3 – step-4, till no new spanning trees from $\mathcal{G}$.

    Step-4: Return $\mathbf{T}$ as set of all spanning trees.

Graph connectedness

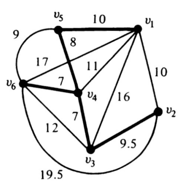
- Spanning tree:
  - Property:: Central tree of $\mathcal{G} = (\mathbf{V}, \mathbf{E})$: spanning tree $\mathcal{T}_0$, such that $d_{\max}(\mathcal{T}_0, \mathcal{T}) \leq \max_{\forall i,j} d(\mathcal{T}_i, \mathcal{T}_j)$, for every tree $\mathcal{T}_i$, $\mathcal{T}_j$ of $\mathcal{G}$, where $d_{\max}(\mathcal{T}_0, \mathcal{T}) =$ maximal distance of $\mathcal{T}_0$ with $\mathcal{T}$ among all other spanning trees of $\mathcal{G}$.

Graph connectedness

- Minimum spanning tree: spanning tree of given weighted undirected connected graph, having minimum total weight among all possible spanning trees of given graph.
  - Property:: Minimum spanning tree of undirected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$: spanning tree $\mathcal{T} = (\mathbf{V}, \mathbf{e})$, $\mathcal{T} \subset \mathcal{G}$, such that $wt(\mathcal{T}) = \min\{wt(\mathcal{T}') \mid \mathcal{T}' =$ one of all possible spanning trees of $\mathcal{G}\}$.
  - Property:: Minimum spanning tree: also called shortest spanning tree, shortest-distance spanning tree, minimal spanning tree.

Graph connectedness

- Minimum spanning tree:
  - Example-1:: Minimum spanning tree:



$$
\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array}
\begin{array}{c}
\begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\
\left[\begin{array}{cccccc}
- & 10 & 16 & 11 & 10 & 17 \\
10 & - & 9.5 & \infty & \infty & 19.5 \\
16 & 9.5 & - & 7 & \infty & 12 \\
11 & \infty & 7 & - & 8 & 7 \\
10 & \infty & \infty & 8 & - & 9 \\
17 & 19.5 & 12 & 7 & 9 & -
\end{array}\right]
\end{array}
$$

[Ref: Narsingh Deo, *Graph Theory with Applications to Engineering and Computer Science*, Prentice-Hall, 1974.]

Graph connectedness

- Minimum spanning tree:
  - Property:: ⟨Theorem⟩: Spanning tree $\mathcal{T}$ of given weighted connected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ to be considered as minimum spanning tree of $\mathcal{G}$, if and only if no other spanning tree $\mathcal{T}'$ of $\mathcal{G}$, at unit distance from $\mathcal{T}$, i.e., $d(\mathcal{T}, \mathcal{T}') = 1$, with weight smaller than that of $\mathcal{T}$ to be present in $\mathcal{G}$.

Graph connectedness

- Minimum spanning tree:
  - Property:: Kruskal's algorithm, Prim's algorithm (also called Prim-Jarník algorithm) to be used to find minimum spanning tree of graph.

Summary

- Focus: Spanning tree.
- Spanning tree, with properties and examples.
- Branch of spanning tree, with properties and examples.
- Chord of spanning tree, with properties and examples.
- Chord set of spanning tree, with properties.
- Rooted spanning tree.
- Spanning forest.
- Weight of spanning tree.
- Spanning tree related theorems, with proofs.
- Rank, nullity of spanning tree.

Summary

- Distance between spanning trees, with properties.
- Fundamental circuit of graph w.r.t. spanning tree , with properties and examples.
- Spanning tree construction algorithms.
- Elementary tree transformation.
- Algorithms to construct all spanning trees of given graph.
- Central tree.
- Minimum spanning tree, with properties and examples.
- Minimum spanning tree related theorems, with proofs.
- Minimum spanning tree construction algorithms.

References

1. [Ros19] Kenneth H. Rosen, *Discrete Mathematics and its Applications*, Eighth edition, McGraw-Hill Education, 2019.
2. [Lip07] Seymour Lipschutz and Marc Lars Lipson, *Schaum's Outline of Theory and Problems of Discrete Mathematics*, Third edition, McGraw-Hill Education, 2007.
3. [Wes01] Douglas Brent West, *Introduction to Graph Theory*, Second edition, Prentice-Hall, 2001.
4. [Deo74] Narsingh Deo, *Graph Theory with Applications to Engineering and Computer Science*, Prentice-Hall, 1974.
5. [Har69] Frank Harary, *Graph Theory*, Addison-Wesley, 1969.