CS34110 Discrete Mathematics and Graph Theory

UNIT – III, Module – 1

Lecture 28: Graphs & Trees

[Tree enumeration; Prüfer sequence; Conversion algorithms of labeled tree ↔ Prüfer sequence; Proof of Cayley's Theorem]

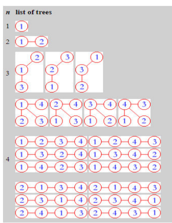
Dr. Suddhasil De

Notation table

Symbol / Notation	Meaning
$\varphi:\mathbb{Z}^{+}\to\mathbf{V}$	Labeling function of labeled tree $\mathcal{T}=(\mathbf{V},\mathbf{E})$.
$\mathbf{T}$	Set of labeled trees formed from vertex set $\mathbf{V}$.
$\mathbb{S}$	Set of Prüfer sequences for encoding trees in $\mathbf{T}$.

Tree enumeration

- Tree enumeration: concerned with counting distinct trees possibly constructed from vertex set of n vertices, where $n \in \mathbb{Z}^{+}$.
  - Property:: For tree enumeration, labeled unrooted trees to be considered.



[Ref: https://www.cs.mcgill.ca/~abatko/computers/free_trees/project/#:~:text=Enumeration%20of%20trees%20is%20concerned,...%20%2C%20n%20%7D.]

Tree enumeration

- Tree enumeration:
  - Property:: Prüfer sequence [also called Prüfer code or Prüfer numbers] of labeled tree $\mathcal{T} = (\mathbf{V}, \mathbf{E})$ ($\mathbf{V} = \{v_1, v_2, \ldots, v_n\}$, $|\mathbf{V}| = n, n \geq 2$): unique sequence $\boldsymbol{S}$ of $(n-2)$ terms (representing encoding of $\mathcal{T}$), with each term representing label of vertex from $\mathbf{V}$, based on labeling function $\varphi: \mathbb{Z}^+ \rightarrow \mathbf{V}$ with $v_1 = \varphi(1)$, $v_2 = \varphi(2), \ldots, v_n = \varphi(n)$, establishing one-to-one correspondence (i.e., bijection) between 'labeled tree $\mathcal{T}$ on n vertices' and 'sequence $\boldsymbol{S}$ of length $(n-2)$ on vertex labels $1, \ldots, n$ associated with $\mathcal{T}$.'

Tree enumeration

- Tree enumeration:
  - Algorithm:: ⟨Steps to convert labeled tree into Prüfer sequence⟩:

Input: $\mathcal{T} = (\mathbf{V}, \mathbf{E})$ ($|\mathbf{V}| = n, n \geq 2$), label in $\mathbf{V}$: $1, \ldots, n$ (as per $\varphi: \mathbb{Z}^+ \rightarrow \mathbf{V}$).
[i.e., vertex set to become vertex sequence V, similar to array]
Output: Sequence $\boldsymbol{S} = \{s_{n-2}\}$.
Step-1: Repeat steps 2-5 for $i = 1, \ldots, (n-2)$.
Step-2: Find pendant vertex with smallest label in V, and record in v_l.
Step-3: $V \leftarrow V \setminus v_l$ (deletion operation).
Step-4: Find neighbor vertex $u_{l'}$ of v_l, having label l', based on *Adj*.
Step-5: $s_i \leftarrow l'$.

Tree enumeration

- Tree enumeration:
  - Algorithm:: ⟨Steps to convert Prüfer sequence into labeled tree⟩:

Input: Sequence $\boldsymbol{S} = \{s_{n-2}\}$, $\varphi(1), \varphi(2), \ldots, \varphi(n)$.
Output: $\mathcal{T} = (\mathbf{V}, \mathbf{E})$ ($|\mathbf{V}| = n, n \geq 2$), adjacency matrix *Adj* of $\mathcal{T}$.
Step-1: Set $\mathbf{V}$ (such that $|\mathbf{V}| = n$): $\mathbf{V} = \{\varphi(1), \varphi(2), \ldots, \varphi(n)\}$.
Step-2: For each vertex v_l in $\mathbf{V}$, set $\deg(v_l)$ = (number of times v_l present in $\boldsymbol{S}$) plus 1.
Step-3: For each term s in $\boldsymbol{S}$, find first smallest-labeled pendant vertex $v_{l'}$ in $\mathbf{V}$ (i.e., $\deg(v_{l'}) = 1$), add edge $e = \{v_{l'}, s\}$ to $\mathbf{E}$, and $\deg(v_{l'}) = \deg(v_{l'}) - 1$. [iteration nesting]

(contd. to next slide)

Tree enumeration

- Tree enumeration:
  - Algorithm:: ⟨Prüfer sequence into labeled tree⟩: contd.

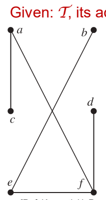
    Step-4: Add edge of two remaining vertices of degree 1 to $\mathbf{E}$, and decrement their degree by 1.

    Step-5: Construct $\mathcal{Adj}$ of $\mathcal{T}$ from $\mathbf{V}$, $\mathbf{E}$, and draw $\mathcal{T} = (\mathbf{V}, \mathbf{E})$.

Tree enumeration

- Tree enumeration:
  - Algorithm:: ⟨Labeled tree into Prüfer sequence⟩:: example-1.

    Given: $\mathcal{T}$, its adjacency matrix $\mathcal{Adj}$, $\mathbf{V}$, $\mathcal{V}$.



$$\mathcal{Adj} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

$\mathbf{V} = \{a, b, c, d, e, f\}$. Iterations $i = 1, \dots, 4$.

$\mathcal{V}$:

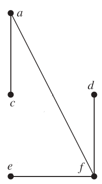
$\varphi(1) = a$	$\varphi(2) = b$	$\varphi(3) = c$	$\varphi(4) = d$	$\varphi(5) = e$	$\varphi(6) = f$
$\deg(a) = 2$	$\deg(b) = 1$	$\deg(c) = 1$	$\deg(d) = 1$	$\deg(e) = 2$	$\deg(f) = 3$
1	2	3	4	5	6

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

Tree enumeration

- Tree enumeration:
  - Algorithm:: ⟨Labeled tree into Prüfer sequence⟩:: example-1 (contd.).

    Iteration $i = 1$:



$$\mathcal{Adj} = \begin{bmatrix} 0 & \cancel{0} & 1 & 0 & 0 & 1 \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \mathbf{X} & \cancel{0} \\ 1 & \cancel{0} & 0 & 0 & 0 & 0 \\ 0 & \cancel{0} & 0 & 0 & 0 & 1 \\ 0 & \mathbf{X} & 0 & 0 & 0 & 1 \\ 1 & \cancel{0} & 0 & 1 & 1 & 0 \end{bmatrix}$$ Vertex satisfying condition: b.

$\mathbf{V} = \{a, \cancel{b}, c, d, e, f\}$. $\mathcal{S} = \{5\}$.

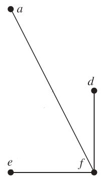
$\mathcal{V}$:

$\varphi(1) = a$	$\varphi(3) = c$	$\varphi(4) = d$	$\varphi(5) = e$	$\varphi(6) = f$
$\deg(a) = 2$	$\deg(c) = 1$	$\deg(d) = 1$	$\deg(e) = 1$	$\deg(f) = 3$
1	2	3	4	5

Tree enumeration

- Tree enumeration:
  - Algorithm:: ⟨Labeled tree into Prüfer sequence⟩:: example-1 (contd-2.).

Iteration $i = 2$:



$$Adj = \begin{bmatrix} 0 & \cancel{0} & \cancel{X} & 0 & 0 & 1 \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & X & \cancel{0} \\ \cancel{X} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ 0 & \cancel{0} & 0 & 0 & 0 & 1 \\ 0 & X & \cancel{0} & 0 & 0 & 1 \\ 1 & \cancel{0} & \cancel{0} & 1 & 1 & 0 \end{bmatrix}$$ Vertex satisfying condition: c.

$\mathbf{V} = \{a, \cancel{b}, \cancel{c}, d, e, f\}$. $\boldsymbol{S} = \{5,1\}$.

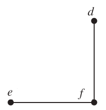
V:

$\varphi(1) = a$ $\deg(a) = 1$	$\varphi(4) = d$ $\deg(d) = 1$	$\varphi(5) = e$ $\deg(e) = 1$	$\varphi(6) = f$ $\deg(f) = 3$
1	2	3	4

Tree enumeration

- Tree enumeration:
  - Algorithm:: ⟨Labeled tree into Prüfer sequence⟩:: example-1 (contd-3.).

Iteration $i = 3$:



$$Adj = \begin{bmatrix} \cancel{0} & \cancel{0} & \cancel{X} & \cancel{0} & \cancel{0} & \cancel{X} \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & X & \cancel{0} \\ \cancel{X} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ \cancel{0} & \cancel{0} & 0 & 0 & 0 & 1 \\ \cancel{0} & X & \cancel{0} & 0 & 0 & 1 \\ \cancel{X} & \cancel{0} & \cancel{0} & 1 & 1 & 0 \end{bmatrix}$$ Vertex satisfying condition: a.

$\mathbf{V} = \{\cancel{a}, \cancel{b}, \cancel{c}, d, e, f\}$. $\boldsymbol{S} = \{5,1,6\}$.

V:

$\varphi(4) = d$ $\deg(d) = 1$	$\varphi(5) = e$ $\deg(e) = 1$	$\varphi(6) = f$ $\deg(f) = 2$
1	2	3

Tree enumeration

- Tree enumeration:
  - Algorithm:: ⟨Labeled tree into Prüfer sequence⟩:: example-1 (contd-4.).

Iteration $i = 4$:

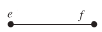
$$Adj = \begin{bmatrix} \cancel{0} & \cancel{0} & \cancel{X} & \cancel{0} & \cancel{0} & \cancel{X} \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & X & \cancel{0} \\ \cancel{X} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ \cancel{0} & \cancel{0} & 0 & \cancel{0} & \cancel{0} & \cancel{X} \\ \cancel{0} & X & \cancel{0} & \cancel{0} & 0 & 1 \\ \cancel{X} & \cancel{0} & \cancel{0} & \cancel{X} & 1 & 0 \end{bmatrix}$$ Vertex satisfying condition: d.

$\mathbf{V} = \{\cancel{a}, \cancel{b}, \cancel{c}, \cancel{d}, e, f\}$. $\boxed{\boldsymbol{S} = \{5,1,6,6\}.}$

Prüfer sequence

V:

$\varphi(5) = e$ $\deg(e) = 1$	$\varphi(6) = f$ $\deg(f) = 1$
1	2



Tree enumeration

- Tree enumeration:
  - Algorithm:: ⟨Prüfer sequence into labeled tree⟩: example-2.
    Given: Prüfer sequence $\boldsymbol{S}$, $\varphi(1), \varphi(2), \ldots, \varphi(n)$.

$\boldsymbol{S} = \{5,1,6,6\}$.

Tree enumeration

- Tree enumeration:
  - Algorithm:: ⟨Prüfer sequence into labeled tree⟩: example-2 (contd.).
    Initialize $\mathbf{V}$, vertex array V with degree 1 for all vertices; $|\mathbf{V}| = |\boldsymbol{S}| + 2$.

$\boldsymbol{S} = \{5,1,6,6\}$.
$\mathbf{V} = \{\varphi(1), \varphi(2), \varphi(3), \varphi(4), \varphi(5), \varphi(6)\}$.

V:

V:	$\varphi(1)$ deg(1) = 1	$\varphi(2)$ deg(1) = 1	$\varphi(3)$ deg(3) = 1	$\varphi(4)$ deg(4) = 1	$\varphi(5)$ deg(5) = 1	$\varphi(6)$ deg(6) = 1
	1	2	3	4	5	6

Tree enumeration

- Tree enumeration:
  - Algorithm:: ⟨Prüfer sequence into labeled tree⟩: example-2 (contd-2.).
    Update vertex array V with degree values as per terms of $\boldsymbol{S}$.

Iterations: outer – $i = 1, \ldots, 4$; inner – $j = 1, \ldots, 6$.
$\boldsymbol{S} = \{5,1,6,6\}$.
$\mathbf{V} = \{\varphi(1), \varphi(2), \varphi(3), \varphi(4), \varphi(5), \varphi(6)\}$.

V:	$\varphi(1)$ deg(1) = 2	$\varphi(2)$ deg(1) = 1	$\varphi(3)$ deg(3) = 1	$\varphi(4)$ deg(4) = 1	$\varphi(5)$ deg(5) = 2	$\varphi(6)$ deg(6) = 3
	1	2	3	4	5	6

Tree enumeration

- Tree enumeration:
  - Algorithm:: ⟨Prüfer sequence into labeled tree⟩: example-2 (contd-3.).

Iteration $i = 1, j = 2$

$\mathbf{E} = \{\{\varphi(2), \varphi(s_1{=}5)\}\}.$

$S = \{\cancel{5}, 1, 6, 6\}.$

$\mathbf{V} = \{\varphi(1), \varphi(2), \varphi(3), \varphi(4), \varphi(5), \varphi(6)\}.$

V:	$\varphi(1)$	$\varphi(2)$	$\varphi(3)$	$\varphi(4)$	$\varphi(5)$	$\varphi(6)$
	deg(1) =2	deg(1) =0	deg(3) =1	deg(4) =1	deg(5) =1	deg(6) =3
	1	2	3	4	5	6

Tree enumeration

- Tree enumeration:
  - Algorithm:: ⟨Prüfer sequence into labeled tree⟩: example-2 (contd-4.).

Iteration $i = 2, j = 3$

$\mathbf{E} = \{\{\varphi(2), \varphi(s_1{=}5)\}, \{\varphi(3), \varphi(s_2{=}1)\}\}.$

$S = \{\cancel{5}, \cancel{1}, 6, 6\}.$

$\mathbf{V} = \{\varphi(1), \varphi(2), \varphi(3), \varphi(4), \varphi(5), \varphi(6)\}.$

V:	$\varphi(1)$	$\varphi(2)$	$\varphi(3)$	$\varphi(4)$	$\varphi(5)$	$\varphi(6)$
	deg(1) =1	deg(1) =0	deg(3) =0	deg(4) =1	deg(5) =1	deg(6) =3
	1	2	3	4	5	6

Tree enumeration

- Tree enumeration:
  - Algorithm:: ⟨Prüfer sequence into labeled tree⟩: example-2 (contd-5.).

Iteration $i = 3, j = 1$

$\mathbf{E} = \{\{\varphi(2), \varphi(s_1{=}5)\}, \{\varphi(3), \varphi(s_2{=}1)\}, \{\varphi(1), \varphi(s_3{=}6)\}\}.$

$S = \{\cancel{5}, \cancel{1}, \cancel{6}, 6\}.$

$\mathbf{V} = \{\varphi(1), \varphi(2), \varphi(3), \varphi(4), \varphi(5), \varphi(6)\}.$

V:	$\varphi(1)$	$\varphi(2)$	$\varphi(3)$	$\varphi(4)$	$\varphi(5)$	$\varphi(6)$
	deg(1) =0	deg(1) =0	deg(3) =0	deg(4) =1	deg(5) =1	deg(6) =2
	1	2	3	4	5	6

Tree enumeration

- Tree enumeration:
  - Algorithm:: ⟨Prüfer sequence into labeled tree⟩: example-2 (contd-6.).

Iteration $i = 4$, $j = 4$

$$\mathbf{E} = \{\{\varphi(2), \varphi(s_1{=}5)\},\ \{\varphi(3), \varphi(s_2{=}1)\},\ \{\varphi(1), \varphi(s_3{=}6)\},\ \{\varphi(4), \varphi(s_4{=}6)\}\}$$

$$\boldsymbol{S} = \{\cancel{5}, \cancel{1}, \cancel{6}, \cancel{6}\}.$$

$$\mathbf{V} = \{\varphi(1), \varphi(2), \varphi(3), \varphi(4), \varphi(5), \varphi(6)\}.$$

V:						
	$\varphi(1)$ deg(1) =0	$\varphi(2)$ deg(1) =0	$\varphi(3)$ deg(3) =0	$\varphi(4)$ deg(4) =0	$\varphi(5)$ deg(5) =1	$\varphi(6)$ deg(6) =1
	1	2	3	4	5	6

Tree enumeration

- Tree enumeration:
  - Algorithm:: ⟨Prüfer sequence into labeled tree⟩: example-2 (contd-7.).

Remaining last edge to be added.

$$\mathbf{E} = \{\{\varphi(2), \varphi(s_1{=}5)\},\ \{\varphi(3), \varphi(s_2{=}1)\},\ \{\varphi(1), \varphi(s_3{=}6)\},\ \{\varphi(4), \varphi(s_4{=}6)\},\ \{\varphi(5), \varphi(6)\}\}$$

$$\boldsymbol{S} = \{\cancel{5}, \cancel{1}, \cancel{6}, \cancel{6}\}.$$

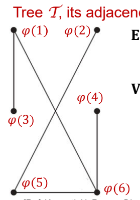
$$\mathbf{V} = \{\varphi(1), \varphi(2), \varphi(3), \varphi(4), \varphi(5), \varphi(6)\}.$$

V:						
	$\varphi(1)$ deg(1) =0	$\varphi(2)$ deg(1) =0	$\varphi(3)$ deg(3) =0	$\varphi(4)$ deg(4) =0	$\varphi(5)$ deg(5) =0	$\varphi(6)$ deg(6) =0
	1	2	3	4	5	6

Tree enumeration

- Tree enumeration:
  - Algorithm:: ⟨Prüfer sequence into labeled tree⟩: example-2 (contd-8.).

Tree $\mathcal{T}$, its adjacency matrix $\mathcal{A}dj$. $\boldsymbol{S} = \{\cancel{5}, \cancel{1}, \cancel{6}, \cancel{6}\}.$



$$\mathbf{E} = \{\{\varphi(2), \varphi(s_1{=}5)\},\ \{\varphi(3), \varphi(s_2{=}1)\},\ \{\varphi(1), \varphi(s_3{=}6)\},\ \{\varphi(4), \varphi(s_4{=}6)\},\ \{\varphi(5), \varphi(6)\}\}$$

$$\mathbf{V} = \{\varphi(1), \varphi(2), \varphi(3), \varphi(4), \varphi(5), \varphi(6)\}$$

$$\mathcal{A}dj = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

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

Tree enumeration

- Tree enumeration:
  - Property:: ⟨Cayley's Theorem⟩: Number of labeled trees to be equal to n^{n-2}, where each labeled tree $\mathcal{T}$ to have n vertices in vertex set $\mathbf{V}$ (i.e., $|\mathbf{V}| = n, n \geq 2$).

  Proof [Prüfer(1918)]: Based on principle of mathematical induction and forming set of Prüfer sequences for trees with n vertices in $\mathbf{V}$.
  To prove: if presence of bijection function $\Phi: \mathbb{T} \to \mathbb{S}$ between $\mathbb{T}$ = set of labeled trees formed from $\mathbf{V}$, and $\mathbb{S}$ = set of Prüfer sequences $\boldsymbol{S}$ of length $(|\mathbf{V}| - 2) = n - 2$ for encoding trees in $\mathbb{T}$. Then, number of labeled trees from $\mathbf{V}$ = unique encoding count = $|\mathbb{S}|$.

(contd. to next slide)

Tree enumeration

- Tree enumeration:

  Proof [Prüfer(1918)] (contd.):
  For establishing bijective nature of $\Phi: \mathbb{T} \to \mathbb{S}$, to show that for each Prüfer sequence $\boldsymbol{S} = \{s_{n-2}\} \in \mathbb{S}$, exactly one tree $\mathcal{T}$ with vertex set $\mathbf{V}$ to generate $\boldsymbol{S}$, i.e., PrüferCoding$(\mathcal{T}) = \boldsymbol{S}$ to have exactly one solution.
  Basis step: $n = 2$. For $\mathcal{T}$ with $|\mathbf{V}| = 2$ (only one tree possible), length of $\boldsymbol{S} = 0$. Then PrüferCoding$(\mathcal{T}) = \boldsymbol{S}$ with exactly one solution.
  Inductive step:
  Inductive hypothesis: premise that unique Prüfer sequence $\boldsymbol{S}'$ for fewer than n vertices, i.e., length of $\boldsymbol{S}'$ less than $(n-2)$.

(contd. to next slide)

Tree enumeration

- Tree enumeration:

  Proof [Prüfer(1918)] (contd-2.):
  Considering unique Prüfer sequence $\boldsymbol{S}'$ formed for tree $\mathcal{T}'$ with $\mathbf{V}'$, $|\mathbf{V}'| = n - 1$, then, PrüferCoding$(\mathcal{T}') = \boldsymbol{S}'$, with length of $\boldsymbol{S}' = n - 3$.
  Main role of algorithmic steps to determine $\boldsymbol{S}'$: to reduce degree of each vertex of $\mathcal{T}'$ to 1 and then possibly deleting it.
  Thus, no pendant vertex v of $\mathcal{T}'$ to appear in $\boldsymbol{S}'$, due to restrictive precondition of reducing $\mathcal{T}'$ to single vertex u before recording v as neighbor of u.
  Hence, every non-pendant vertex of $\mathcal{T}'$ to appear in $\boldsymbol{S}'$.

(contd. to next slide)

Tree enumeration

- Tree enumeration:
  Proof [Prüfer(1918)] (contd-3.):
  For any labeled tree $\mathcal{T} = (\mathbf{V}, \mathbf{E})$, $|\mathbf{V}| = n$, let encoding of $\mathcal{T}$ resulting into $\boldsymbol{S}$, i.e., PrüferCoding$(\mathcal{T}) = \boldsymbol{S}$, where length of $\boldsymbol{S} = n - 2$.
  To show that $\boldsymbol{S}$ unique for $\mathcal{T}$, i.e., exactly one solution to above.
  For same reasoning applied to $\mathcal{T}'$ and $\boldsymbol{S}'$ (in last slide), every non-pendant vertex of $\mathcal{T}$ to appear in $\boldsymbol{S}$, and no pendant vertex of $\mathcal{T}$ in $\boldsymbol{S}$.
  Let pendant vertex $v' \in \mathbf{V}$ of $\mathcal{T}$ to be deleted first, then v' to have smallest label in $\mathbf{V}$, and not to appear in $\boldsymbol{S}$.

(contd. to next slide)

Tree enumeration

- Tree enumeration:
  Proof [Prüfer(1918)] (contd-4.):
  Further, $\mathcal{T} - v' = \mathcal{T}'$, such that $\mathbf{V}' = \mathbf{V} \setminus \{v'\}$, and then $|\mathbf{V}'| = n - 1$.
  So, deleting v' resulting into converting $\mathcal{T}$ to become $\mathcal{T}'$.
  As per inductive hypothesis, exactly one tree $\mathcal{T}'$ with vertex set $\mathbf{V}'$ and Prüfer sequence $\boldsymbol{S}'$.
  Let non-pendant vertex $u' \in \mathbf{V}$ be neighbor of v' in $\mathcal{T}$, through edge $e = \{v', u'\}$, $e \in \mathbf{E}$.
  Above TRUE for every labeled tree, with least vertex v' and edge $\{v', u'\}$.

(contd. to next slide)

Tree enumeration

- Tree enumeration:
  Proof [Prüfer(1918)] (contd-5.):
  Since every tree with Prüfer sequence $\boldsymbol{S}$ formed by adding edge $\{v', u'\}$ to $\mathcal{T}'$, so at most one solution to: PrüferCoding$(\mathcal{T}) = \boldsymbol{S}$.
  Again, adding $\{v', u'\}$ to $\mathcal{T}'$ indeed creating tree with vertex set $\mathbf{V}$ and Prüfer sequence $\boldsymbol{S}$, so at least one solution to: PrüferCoding$(\mathcal{T}) = \boldsymbol{S}$.
  Hence, exactly one solution to PrüferCoding$(\mathcal{T}) = \boldsymbol{S}$, where $\boldsymbol{S} \in \mathbb{S}$.
  Finally, $|\mathbb{S}|$ = number of unique Prüfer sequences (of length $n - 2$) for labeled trees with n vertices = number of $(n - 2)$-permutations with repetitions from n labels $= n^{n-2}$. ∎

Summary

- Focus: Tree enumeration.
- Tree enumeration.
- Prüfer sequence, with properties.
- Algorithm to convert labeled tree into Prüfer sequence.
- Algorithm to convert Prüfer sequence into labeled tree.
- Labeled tree into Prüfer sequence detailed steps, with examples.
- Prüfer sequence into labeled tree detailed steps, with examples.
- Rigorous proof of Cayley’s Theorem using Prüfer sequences.

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.
6. https://en.wikipedia.org/wiki/Prüfer_sequence.