Lecture 4. Cayley's formula and Prüfer code

Yanbo ZHANG

Hebei Normal University

Question

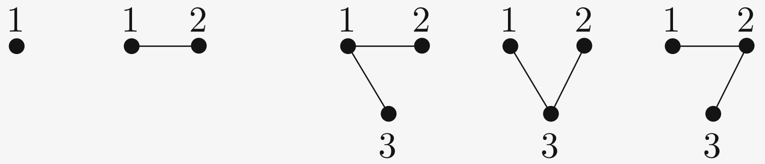
Question

What is the number of spanning trees in a labelled complete graph on n vertices?

Question

Question

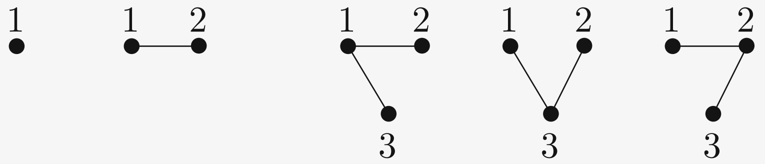
What is the number of spanning trees in a labelled complete graph on n vertices?



Question

Question

What is the number of spanning trees in a labelled complete graph on n vertices?

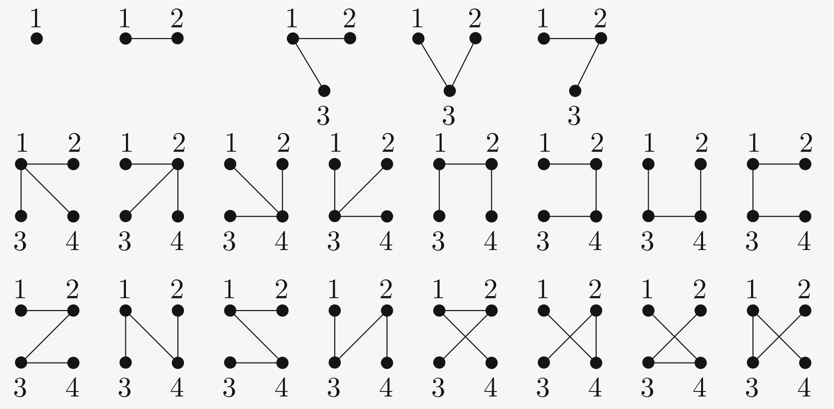


The same question is:

Question

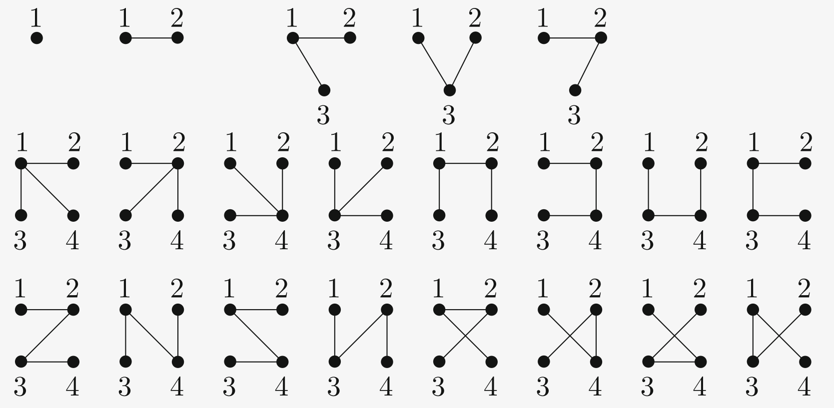
How many different trees can we get on the vertex set $N = \{1, 2, \ldots, n\}$?

Labelled trees of small order



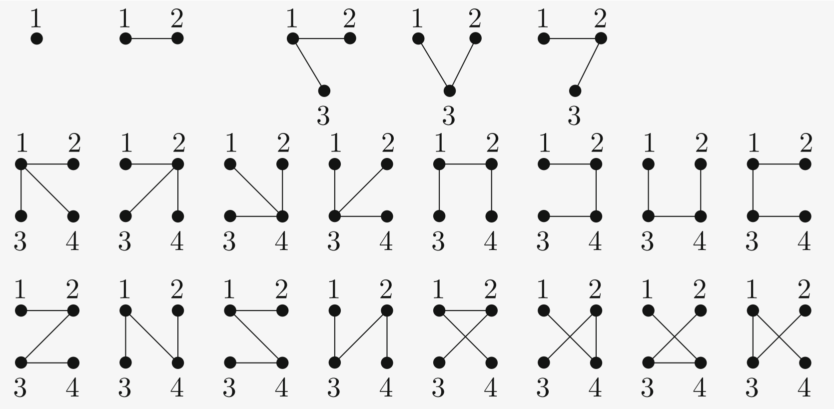
Enumeration "by hand" yields $T_1 = 1$, $T_2 = 1$, $T_3 = 3$,

Labelled trees of small order



Enumeration "by hand" yields $T_1 = 1$, $T_2 = 1$, $T_3 = 3$, $T_4 = 16$.

Labelled trees of small order

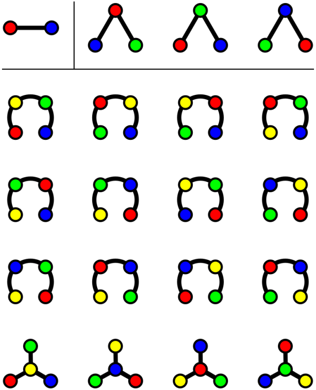


Enumeration "by hand" yields $T_1 = 1$, $T_2 = 1$, $T_3 = 3$, $T_4 = 16$.

Conjecture that $T_n = n^{n-2}$

Theorem (Cayley's formula)

There are n^{n-2} trees with vertex set $[n]$.



Prüfer code

To prove Cayley's formula, the most direct method is to find a bijection from the set of all trees on n vertices onto another set whose cardinality is known to be n^{n-2}.

Prüfer code

To prove Cayley's formula, the most direct method is to find a bijection from the set of all trees on n vertices onto another set whose cardinality is known to be n^{n-2}. So we consider the set of all ordered sequences

$$(a_1, a_2, \ldots, a_{n-2}) \text{ with } 1 \le a_i \le n$$

Prüfer code

To prove Cayley's formula, the most direct method is to find a bijection from the set of all trees on n vertices onto another set whose cardinality is known to be n^{n-2}. So we consider the set of all ordered sequences

$$(a_1, a_2, \ldots, a_{n-2}) \text{ with } 1 \leq a_i \leq n$$

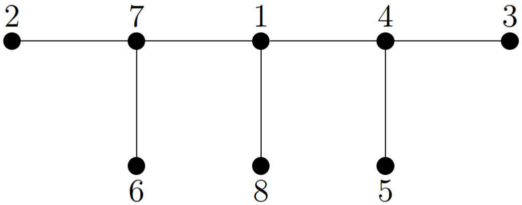
Definition (Prüfer code)

Let T be a tree on an ordered set S of n vertices. To compute the Prüfer sequence $f(T)$, iteratively delete the leaf with the smallest label and append the label of its neighbour to the sequence. After $n-2$ iterations a single edge remains and we have produced a sequence $f(T)$ of length $n-2$.

Get a Prüfer code from a tree

Iteratively delete the leaf with the smallest label and append the label of its neighbour to the sequence.

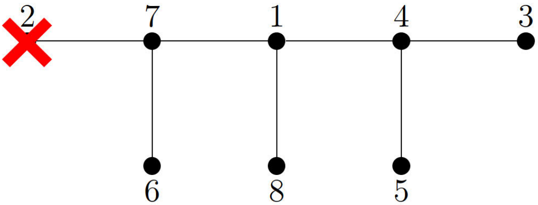
$T =$



$$f(T) = \emptyset$$

Iteratively delete the leaf with the smallest label and append the label of its neighbour to the sequence.

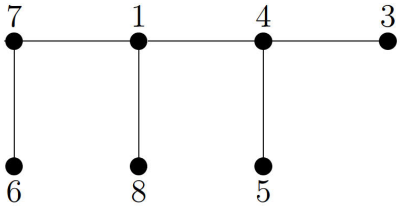
$T =$



$$f(T) = \emptyset$$

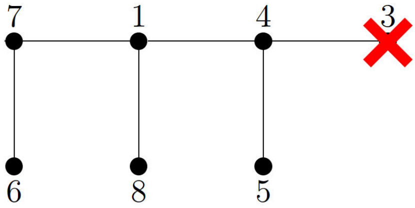
Get a Prüfer code from a tree

Iteratively delete the leaf with the smallest label and append the label of its neighbour to the sequence.



$$f(T) = (7)$$

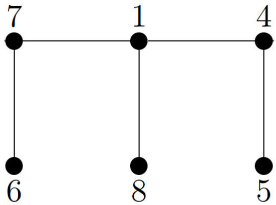
Iteratively delete the leaf with the smallest label and append the label of its neighbour to the sequence.



$$f(T) = (7)$$

Get a Prüfer code from a tree

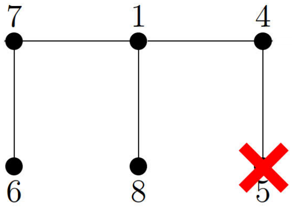
Iteratively delete the leaf with the smallest label and append the label of its neighbour to the sequence.



$$f(T) = (7, 4)$$

Get a Prüfer code from a tree

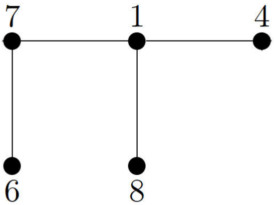
Iteratively delete the leaf with the smallest label and append the label of its neighbour to the sequence.



$$f(T) = (7, 4)$$

Get a Prüfer code from a tree

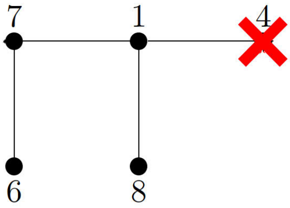
Iteratively delete the leaf with the smallest label and append the label of its neighbour to the sequence.



$$f(T) = (7, 4, 4)$$

Get a Prüfer code from a tree

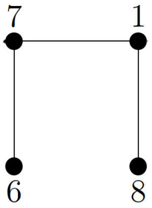
Iteratively delete the leaf with the smallest label and append the label of its neighbour to the sequence.



$$f(T) = (7,4,4)$$

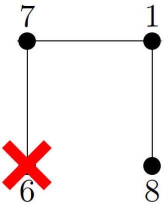
Get a Prüfer code from a tree

Iteratively delete the leaf with the smallest label and append the label of its neighbour to the sequence.



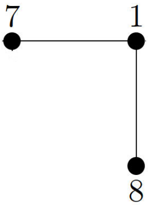
$$f(T) = (7, 4, 4, 1)$$

Iteratively delete the leaf with the smallest label and append the label of its neighbour to the sequence.



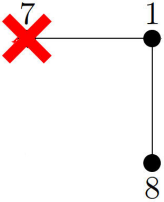
$$f(T) = (7,4,4,1)$$

Iteratively delete the leaf with the smallest label and append the label of its neighbour to the sequence.



$$f(T) = (7, 4, 4, 1, 7)$$

Iteratively delete the leaf with the smallest label and append the label of its neighbour to the sequence.



$$f(T) = (7, 4, 4, 1, 7)$$

Iteratively delete the leaf with the smallest label and append the label of its neighbour to the sequence.

$$f(T) = (7, 4, 4, 1, 7, 1)$$

Get a Prüfer code from a tree

Iteratively delete the leaf with the smallest label and append the label of its neighbour to the sequence.

$$f(T) = (7,4,4,1,7,1)$$

After $n-2$ iterations a single edge remains and we have produced the sequence $f(T)$ of length $n-2$.

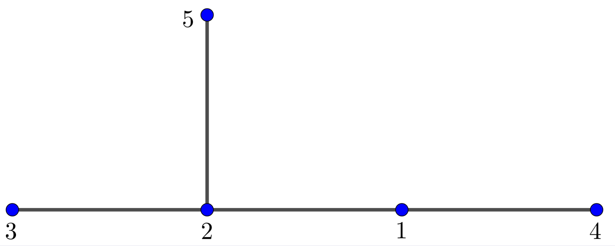
Get a Prüfer code from a tree

Iteratively delete the leaf with the smallest label and append the label of its neighbour to the sequence.

Iteratively delete the leaf with the smallest label and append the label of its neighbour to the sequence.

Question

Find the Prüfer sequence $f(T)$ of T, where T is the following tree:



Get a tree from a Prüfer code

Question

Get a labelled tree T from a Prüfer sequence S.

$$S = (a_1, a_2, \ldots, a_{n-2}) \text{ with } 1 \le a_i \le n$$

At each step,

- (a) Find the smallest unmarked element x not in S;
- (b) join x to the first element of S, delete the first element from S and mark x.

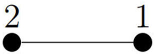
Get a tree from a Prüfer code

Example

Compute the tree with Prüfer code 16631.

$S = 16631$, the unmarked elements are $\{1, 2, 3, 4, 5, 6, 7\}$.

- (a) The smallest unmarked element not in S is 2;
- (b) join 2 to 1, delete 1 from S and mark 2.



Get a tree from a Prüfer code

Example

Compute the tree with Prüfer code 16631.

$S = 6631$, the unmarked elements are $\{1, 3, 4, 5, 6, 7\}$.

- (a) The smallest unmarked element not in S is 4;
- (b) join 4 to 6, delete 6 from S and mark 4.

Get a tree from a Prüfer code

Example

Compute the tree with Prüfer code 16631.

$S = 631$, the unmarked elements are $\{1, 3, 5, 6, 7\}$.

- (a) The smallest unmarked element not in S is 5;
- (b) join 5 to 6, delete 6 from S and mark 5.

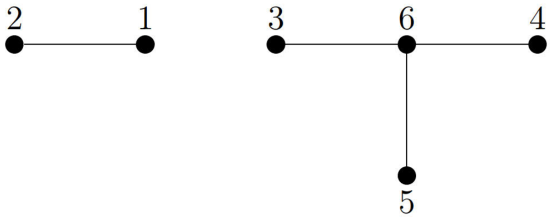
Get a tree from a Prüfer code

Example

Compute the tree with Prüfer code 16631.

$S = 31$, the unmarked elements are $\{1,3,6,7\}$.

- (a) The smallest unmarked element not in S is 6;
- (b) join 6 to 3, delete 3 from S and mark 6.



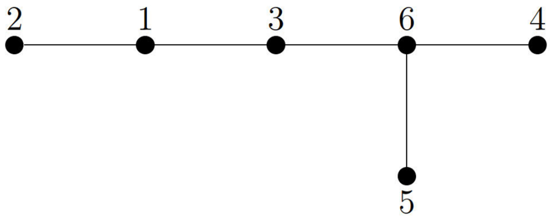
Get a tree from a Prüfer code

Example

Compute the tree with Prüfer code 16631.

$S = 1$, the unmarked elements are $\{1, 3, 7\}$.

- (a) The smallest unmarked element not in S is 3;
- (b) join 3 to 1, delete 1 from S and mark 3.

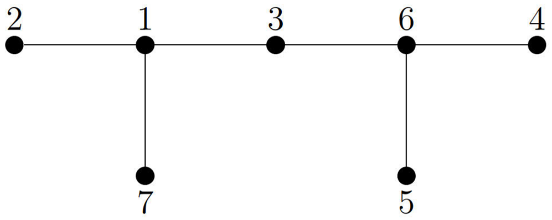


Example

Compute the tree with Prüfer code 16631.

$S = \emptyset$, the unmarked elements are $\{1, 7\}$.

Now add an edge between the remaining vertices $\{1, 7\}$.



At each step,

- (a) Find the smallest unmarked element x not in S;
- (b) join x to the first element of S, delete the first element from S and mark x.

At each step,

- (a) Find the smallest unmarked element x not in S;
- (b) join x to the first element of S, delete the first element from S and mark x.

Question

A tree T has Prüfer code $f(T) = (1,1,3,1)$. Draw this tree.

At each step,

- (a) Find the smallest unmarked element x not in S;
- (b) join x to the first element of S, delete the first element from S and mark x.

Question

A tree T has Prüfer code $f(T) = (2,2,3)$. Write down all the leaves of this tree.

Proposition

For an ordered n-element set S, the Prüfer code f is a bijection between the trees with vertex set S and the sequences in S^{n-2}.

Bijection between a tree and a Prüfer code

Proposition

For an ordered n-element set S, the Prüfer code f is a bijection between the trees with vertex set S and the sequences in S^{n-2}.

Proof.

We need to show every sequence $(a_1,a_2,\ldots,a_{n-2}) \in S^{n-2}$ defines a unique tree T such that $f(T) = (a_1,a_2,\ldots,a_{n-2})$. If $n = 2$, then there is exactly one tree on 2 vertices and the algorithm defining f always outputs the empty sequence, the only sequence of length zero. So the claim clearly holds for $n = 2$.

Now, assume $n > 2$ and the claim holds for all ordered vertex sets S' of size less than n. Consider a sequence $(a_1,a_2,\ldots,a_{n-2}) \in S^{n-2}$. We need to show that $(a_1,a_2,\ldots,a_{n-2})$ can be uniquely produced by the algorithm.

Bijection between a tree and a Prüfer code

Proposition

For an ordered n-element set S, the Prüfer code f is a bijection between the trees with vertex set S and the sequences in S^{n-2}.

Proof.

(continued) Suppose that the algorithm produces $f(T) = (a_1, a_2, \ldots, a_{n-2})$ for some tree T. Then the vertices $\{a_1, a_2, \ldots, a_{n-2}\}$ are precisely those that are not a leaf in T. Indeed, if a vertex v is a leaf in T then it can only appear in $f(T)$ if its neighbour gets deleted during the algorithm. But this would leave v as an isolated vertex, which is impossible. Conversely, if a vertex v is not a leaf then one of its neighbours must be deleted during the algorithm (it cannot be itself deleted before this happens). When this neighbour of v is deleted, v will be added to the Prüfer code for T, so is in $\{a_1, a_2, \ldots, a_{n-2}\}$.

Proposition

For an ordered n-element set S, the Prüfer code f is a bijection between the trees with vertex set S and the sequences in S^{n-2}.

Proof.

(continued) This implies that the label of the first leaf removed from T is the minimum element of the set $S \setminus \{a_1, a_2, \ldots, a_{n-2}\}$. Let v be this element. In other words, in every T such that $f(T) = (a_1, a_2, \ldots, a_{n-2})$ the vertex v is a leaf whose unique neighbour is a_1.

By induction, there is a unique tree T' with vertex set $S \setminus v$ such that $f(T') = (a_2, \ldots, a_{n-2})$. Adding the vertex v and the edge (a_1, v) to T' yields the desired unique tree T with $f(T) = (a_1, a_2, \ldots, a_{n-2})$. □

Thank you!