Number of labeled trees and Prufer Sequences

Sunil Sitaram Shelke

Abstract

A Prufer sequence is a unique sequence of a tree. Prufer sequences are used to count number of labeled trees on n nodes. Once we count the number of Prufer sequences and establish bijection between set of Prufer sequences of certain length and set of labeled trees, we directly count number of labeled trees.

1 Prufer sequence of a labeled tree

A labeled tree on n nodes is an acyclic graph having nodes distinguishable from each other via labels attached to nodes. Consider, trees shown in below figure:

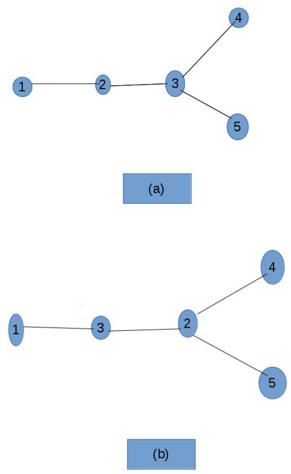

Figure 1: labeled trees

Though, both trees have same Y structure visually, these are different trees because of of permutations of labels 2 and 3 in trees. Counting number of such labeled trees is somehow simpler than counting unlabeled trees. Consider a tree T with node label set $\{1, 2, \ldots, n\}$. A Prufer sequence $P(T)$ of T is constructed through below mentioned procedure:

1. $i = 1$

2. Find the smallest labeled node, say v_k, having degree 1. Delete v_k from T and write it's neighbor node label as i^{th} element of sequence $P(T)$.

3. Follow step 2 as long as there remain more than 2 nodes in the tree.

For eq. Prufer sequence of the tree in fig.1(a) is $2-3-3$ and Prufer sequence of tree in (b) is $3-2-2$. Basically, for n node tree, we construct an $n-2$ length Prufer sequence. Consider, set P_{n-2} of all all possible $n-2$ length sequences over the given fixed label set of tree nodes. Consider $T(n)$ to be the set of all possible n-node labeled trees. There is a bijection $f: P_{n-2} \to T(n)$. Bijection is as follows:-

1. *Claim: For every tree T, Prufer sequence is unique.*
   At each step, node of degree 1 is deleted. Multiple nodes might have degree 1. We resolve uncertainty by deleting a node having smallest label among all degree 1 vertices, say x. So, uniqueness of sequence depends on existence of unique neighbor of x. Every degree-1 node has a unique neighbor. How do we make sure that we always have a degree 1 vertex to delete? Say, at some point of time all remaining nodes have degree $\geq$ 2. This directly implies that remaining graph is *cyclic* and hence not a tree. But since we started with a tree, some vertex has to ave degree 1. Hence such a vertex will have unique neighbor and if there are multiple such nodes then least labeled node is also unique since no 2 nodes have same label. Hence, at each step a unique label is added to partial Prufer sequence. Hence Prufer sequence, as a whole is also unique.

2. *Claim: Given a Prufer sequence P, a unique tree T is constructed.*
   Consider the below procedure to construct a tree T, given Prufer sequence P :

   Let $P = (l_1, l_2, , l_{n2})$ be a Prfer sequence. Initially T contains labeled nodes without any edges.

   (a) Make a set of all the integers $1, 2, , n$. This is our working set W.
   (b) If there are 2 numbers left in W, connect them with an edge in T and stop. Otherwise, continue on to step 3.
   (c) Find the smallest number in $x \in W$ which is not in the sequence P. Let first element of P be y. Add an edge (x,y) in T. Remove x from W and y from P. Go to Step 2.

   The proof of uniqueness of output of above procedure is almost the same as in while constructing Prufer sequence. x's in W were precisely those nodes of degree 1 which were removed from given tree, in increasing label order. While constructing T from P, we trace back the same sequence by maintaining set of labels which never appeared in P and which disappear from P later, one by one.

3. $|T(n)| = n^{n-2}$
   Consider $P \in P_{n-2}$. $|P| = n-2$. We have label set $L = \{1, 2, \ldots, n\}$. Each position in P can be taken by any one of labels in L with multiple occurrences. Hence $|P_{n-2}| = n^{n-2}$. And since we have a bijection $f : P_{n-2} \to T(n)$, $|T(n)| = n^{n-2}$.

2 Discussion

1. It is fixed that single degree nodes should be deleted. But it is not necessary that nodes be deleted in increasing label order. Even decreasing label order would work. But one should delete single degree nodes only.

2. Prufer sequence of a tree lists neighbors of vertices to be deleted, one by one. What if we sequence given tree not by maintaining sequence of neighbor's label but label of node to be deleted and sequence is of length $n-1$ to make sure longer codes may not lose structural information? Will it be another way to encode trees? Will it even work?
   Creating a sequence of nodes to be deleted will not maintain bijection, since at-least 2 trees will have same sequence. For eq. consider a delete sequence $D =< V_{\rho_1}, V_{\rho_1}, \ldots, V_{\rho_n-1} >$

where ρ is a permutation function on $\{1, 2, \ldots, n\}$. One can readily imagine a tree which is a *chain*. Such a tree would be a chain with leftmost node having first element of D as it's label and last node would have label not present in the sequence D. Tree would be D extended to have one node absent in D.
Consider tree in fig.1(a). This tree has delete sequence $D =< 1, 2, 4, 3 >$. Given tree is not a chain graph. But D combined with $\{3\}$ viewed as a path graph $< 1, 2, 4, 3, 5 >$ is also a tree having same delete sequence. In short delete sequence does not encode tree uniquely because delete sequence itself can be a subtree which would destroy bijection.
This implies that delete sequence is not capable of storing structural information of tree in it. Any code which is capable of storing structural information of a tree either directly/indirectly is good enough to identify a tree uniquely.

3. How does Prufer sequence hold complete structural information of a tree?
   Creation of Prufer sequence has 2 key steps. *(a)* We note neighbor's label in sequence. *(b)* We delete nodes in increasing order of labels. What if we delete in arbitrary order of labels but maintain *(a)*? By looking at first element of sequence, we would'nt know which deleted node was it's neighbor. Deleting nodes in increasing order of labels has following property:

   - Increasing order of label deletion promises that we always have some vertex of degree 1 to be deleted. This is very important since the resulting tree will still be connected and same procedure can be iteratively carried out.

Connectedness throughout is important. If we delete a high degree node then tree will be disconnected into at-least 2 trees, say T_1 and T_2 having say n_1 nodes in T_1 and n_2 nodes in T_2. Once we get neighbor sequence, we will not be sure where to split input neighbor sequence to find out how long prefix sequence encoded T_1 and remaining encoded T_2.
This does not mean that creating a sequence by deleting nodes in high degree to low degree will not work at all. But sequence will not be sufficient. We will require additional information like number of components created after each deletion and labels of nodes in each component.