JOURNAL OF ALGORITHMS 2, 204–207 (1981)

NOTE

The Uniform Selection of Free Trees

HERBERT S. WILF*

Department of Mathematics, University of Pennsylvania, Philadelphia, Pennsylvania 19104

Received March 1, 1981

For a given $n > 0$ we want to choose, uniformly at random (u.a.r), a free tree on n vertices. Many algorithms exist for the random selection of combinatorial objects (see [1]), but for families of unlabeled graphs these are rare. Our approach is to reduce the question to one that is closely related to the Procedure Ranrut(n) for the selection of rooted trees on n vertices [1, Chap. 29]. Perhaps not surprisingly, we use the centroid(s) of the tree to carry out the reduction. The resulting algorithm runs in about the same time as Ranrut itself.

By the *weight* of a vertex v in a tree T we mean the number of vertices in the largest subtree at v. A vertex of minimum weight is a *centroid* of T. The main facts about centroids are [2, p. 387] that

(C1) every tree has one or two centroids;

(C2) if T has two centroids then these are joined by an edge of T, and removal of that edge would leave two trees of equal sizes;

(C3) a vertex X is the unique centroid of a tree of n vertices iff its weight is at most $(n - 1)/2$.

We consider separately the cases where the output tree is to have two centroids and where it will have just one. The full algorithm will follow by choosing between these cases with the correct probabilities.

The Selection of Bi-centroidal Trees

For even n we construct, u.a.r., a free tree with two centroids by choosing two rooted trees of $n/2$ vertices and joining them at their roots by a new edge. The probabilities are slightly skewed, however, because a tree in which

*Research supported by the National Science Foundation.

0196-6774/81/020204-04$02.00/0

the two rooted subtrees are the same will occur only half as often as a tree in which they are different.

This seems to call for a tree-isomorphism test followed by a rejection procedure. However, we can avoid the whole problem by a little trick:

Let N objects $\omega_1, \omega_2, \ldots, \omega_N$ be given. In order to choose, with equal a priori probabilities, one of the $\binom{N+1}{2}$ 2-samples (with replacement) of these objects without ever comparing two of them we can

With probability $1/(N+1)$ **do**:

Choose i, $1 \le i \le N$, u.a.r., and output (ω_i, ω_i)

else do:

Independently choose two objects (ω_i, ω_j) (with replacement) and output them. □

Hence if $t_n^{(2)}$ is the number of bi-centroidal free trees and a_n is the number of rooted trees on n vertices, then we have, by property (C2) of centroids,

$$t_n^{(2)} = \binom{a_{n/2} + 1}{2}$$

and to choose one of them we use

Algorithm Bicenter (n)

[Input is an even $n > 0$; output is a bi-centroidal tree T on n vertices, selected u.a.r.]

With probability $(a_{n/2} + 1)^{-1}$ **do** step B1 **else do** B2:

(B1) Set $T' \leftarrow$ Ranrut$(n/2)$, and
$T \leftarrow$ {Two copies of T', joined by an edge at their roots}
Exit

(B2) Set $T' \leftarrow$ Ranrut$(n/2)$
$T'' \leftarrow$ Ranrut$(n/2)$
$T \leftarrow$ {T', T'' joined by an edge at their roots}
Exit □

The Selection of Trees with One Centroid

This is a little harder. Evidently there is a bijection between free trees with one centroid and rooted trees with one centroid that coincides with the root. We let $t_n^{(1)}$ be the number of these on n vertices.

By property (C3) of centroids (above) the problem becomes the selection u.a.r. of one of the $t_n^{(1)}$ rooted trees on n vertices such that all subtrees at the

root are of size $(n-1)/2$ at most, or equivalently, of one of the $t_n^{(1)}$ rooted forests on $n-1$ vertices such that each connected component contains $\leqq (n-1)/2$ vertices.

More generally, for a fixed integer $q > 0$, consider the set of all rooted forests whose connected components have $\leqq q$ vertices. This set is a *prefab* (see [1, pp. 78–81]) i.e., a combinatorial family in which every object is uniquely constructed from *prime* objects, in this case, from rooted trees of at most q vertices.

Let $\alpha(m, q)$ be the number of rooted forests of m vertices whose trees have at most q vertices each. Then

$$\sum_{m \geqq 0} \alpha(m, q)x^m = \prod_{j=1}^{q} \frac{1}{(1-x^j)^{a_j}} \tag{1}$$

and consequently we can follow the general procedure of [1, p. 81], which in this case becomes

Algorithm Forest (m, q)

[Returns a rooted forest on m vertices, each of whose connected components has $\leqq q$ vertices, selected u.a.r.]

If $m = 0$ **then** exit with the empty forest, **else do**

(F1) Choose a pair of integers (j, d) such that

$$\text{Prob}(j, d) = \frac{d\alpha(m-jd, q)a_d}{m\alpha(m, q)} \qquad (j \geqq 1; 1 \leqq d \leqq q).$$

(F2) Recursively set

$$\mathcal{F}' \leftarrow \text{Forest}(m-jd, q)$$

and let

$$T' \leftarrow \text{Ranrut}(d).$$

Then exit with j copies of T' adjoined to $\mathcal{F}'$. □

The numbers $\alpha(m, q)$ can be calculated from

$$m\alpha(m, q) = \sum_{j \geqq 1} \sum_{d=1}^{q} \alpha(m-jd, q)d\alpha(d-1, q) \qquad (m \geqq 1; \alpha(0, q) = 1). \tag{2}$$

Note that $\alpha(m, q) = a_{m+1}$ $(m \leqq q)$. It is also important that the parameter q does not change on the recursive call in step (F2).

Choosing a Free Tree

Finally, we select a free tree u.a.r. by combining the above two procedures into a single

Algorithm Free (n)

[Returns a free tree T on n vertices, selected u.a.r.]

(T1) If n is odd then **do** $p \leftarrow 0$ **else do** $p \leftarrow \binom{1+a_{n/2}}{2}/a_n$.

(T2) *With probability* p **do**:

[the output tree will have two centroids]

$T \leftarrow$ Bicenter(n); Exit with T.

Else do

[the output tree will have one centroid]

$\mathcal{F} \leftarrow$ Forest $(n-1,(n-1)/2)$

$T \leftarrow$ {a new vertex v, joined to all of the roots of $\mathcal{F}$}

Exit with T. □

REFERENCES

1. A. NIJENHUIS and H. S. WILF, "Combinatorial Algorithms," 2nd ed., Academic Press, New York, 1978.
2. D. E. KNUTH, "The Art of Computer Programming," Vol. 1, Addison-Wesley, Reading, Mass., 1968.