

---

Résumé : “pour générer uniformément, il suffit de savoir dénombrer”.

---

## Exercice 1: Génération d’arbres binaires

Dans cet exercice, on cherche à construire un algorithme afin de tirer aléatoirement un arbre binaire à  $n$  sommets uniformément parmi tous les arbres binaires à  $n$  sommets.

1. On se propose l’algorithme suivant :

```
Entrées : Un nombre  $n$ .  
Sorties : Un arbre binaire.  
1 si  $n = 0$  alors  
2   | Retourner  $\emptyset$   
3 sinon  
4   |  $q \leftarrow$  un nombre uniforme parmi  $\{0, \dots, n - 1\}$ ;  
5   | retourner  $ARBRE(arbre\_aleatoire(q), arbre\_aleatoire(n - q - 1))$ 
```

### Algorithme 1 : $arbre\_aleatoire(n)$

- a) Dessiner tous les arbres de taille  $n \in \{1, 2, 3\}$ .
  - b) Pour chaque arbre, calculer la probabilité que l’algorithme 1 produise cet arbre.
  - c) L’algorithme 1 produit-il une distribution *uniforme* ?
2. On note  $C_n$  le nombre d’arbres binaire à  $n$  sommets.
    - a) Que valent  $C_1, C_2, C_3$  ?
    - b) Exprimer  $C_n$  en fonction des  $C_q$  (pour  $q < n$ ).
    - c) En déduire un algorithme de génération uniforme d’arbres binaires.

## Exercice 2: Permutons-nous

1. Combien y a-t-il de permutations à  $n$  éléments ?
2. En déduire un algorithme pour générer uniformément une permutation à  $n$  éléments.

## Exercice 3: Arbres étiquetés

Dans cet exercice, on cherche à générer uniformément un arbre étiqueté à  $n$  sommets. Un arbre étiqueté est un graphe non orienté connexe et acyclique sur l’ensemble des sommets  $\{1, \dots, n\}$ .

1. Combien y a-t-il d’arbres étiquetés à  $n = 2, n = 3$  ou  $n = 4$  sommets ?
2. On se propose de transformer un arbre en un mot à l’aide de l’algorithme suivant :

```
Entrées : Un arbre  $T$  étiqueté par  $\{1 \dots n\}$   
Sorties : Un mot  
1  $W \leftarrow \varepsilon$  ; /*  $W$  est un mot vide */  
2 tant que  $T$  contient 3 noeuds ou plus faire  
3   | Soit  $x$  la feuille de l’arbre ayant la plus petite étiquette;  
4   |  $W \leftarrow W + \text{Père}(x)$  ; /*  $\text{Père}(x)$  est l’étiquette du seul voisin de  $x$  dans  $T$  */  
5   | Enlever  $x$  de  $T$ .  
6 retourner  $W$ 
```

### Algorithme 2 : $\text{Codage\_Prufer}(T)$

3. Faire tourner l’algorithme sur des arbres de taille  $n = 3$  et 4.
4. Montrer que cet algorithme construit une bijection vers un ensemble de mots que l’on décrira.
5. Calculer le cardinal de cet ensemble de mots et donc le nombre d’arbres étiquetés à  $n$  sommets.
6. En déduire un algorithme de génération uniforme d’arbres étiquetés.