

## Corrigé

1. On va établir une relation de récurrence pour les nombres  $T_N$ . Si  $N \geq 1$ , alors un arbre  $t$  dans  $\mathfrak{T}(N)$  est construit en prenant la racine, et en choisissant une famille de sous-arbres  $t_1, \dots, t_r$  tels que  $|t_1| + |t_2| + \dots + |t_r| = N - 1$  : les  $t_i$  sont les sous-arbres de  $t_i$  issus des fils de la racine de  $t$ . Notons  $R_a$  le nombre de sous-arbres  $t_i$  de taille  $a$ . Le nombre de multi-ensembles (ensembles avec éventuellement des multiplicités) constitués de  $R$  éléments choisis dans un ensemble de taille  $T$  est

$$\binom{T + R - 1}{R}.$$

On a donc :

$$T_N = \sum_{(R_i)_{i \geq 1} \mid R_1 + 2R_2 + \dots + sR_s = N - 1} \binom{T_1 + R_1 - 1}{R_1} \binom{T_2 + R_2 - 1}{R_2} \dots \binom{T_s + R_s - 1}{R_s}.$$

En sommant la série  $\sum_{N=1}^{\infty} T_N x^N$ , on obtient alors :

$$\begin{aligned} \sum_{N \geq 1} T_N x^N &= \sum_{(R_i)_{i \geq 1}, R_i \geq 0} x^{1+R_1+2R_2+\dots} \binom{T_1 + R_1 - 1}{R_1} \binom{T_2 + R_2 - 1}{R_2} \dots \binom{T_s + R_s - 1}{R_s} \\ &= x \prod_{i=1}^{\infty} \left( \sum_{R_i=0}^{\infty} \binom{T_i + R_i - 1}{R_i} (x^i)^{R_i} \right) = x \prod_{i=1}^{\infty} \frac{1}{(1 - x^i)^{T_i}}. \end{aligned}$$

2. Le programme suivant calcule toutes les valeurs de  $T_M$  pour  $M \in [1, N]$  :

```
import math

def binomial(a,b):
    if a >= b:
        return math.factorial(a)/(math.factorial(b) * math.factorial(a-b))
    else:
        return 0

def partitions(N):
    """ Retourne l'ensemble des partitions de N en somme d'entiers. """
    if N == 0:
        return [[]]
    elif N == 1:
        return [[1]]
    else:
        res = [[N]]
        for k in range(1,N):
            for p in partitions(k):
                if p[0] <= N-k:
                    res.append([N-k]+p)
        return res

def multiplicities(p):
    """ Prend une partition d'entiers p et retourne les multiplicités R_i
```

```

    des entiers i dans p, sous forme de dictionnaire. """
res = {}
for i in set(p):
    res[i] = p.count(i)
return res

def NombreArbres(N):
    res = [0,1]
    M = 1
    while M < N:
        M += 1
        S = 0
        for p in partitions(M-1):
            pm = multiplicities(p)
            toadd = 1
            for i in pm.keys():
                toadd *= binomial(res[i]+pm[i]-1, pm[i])
            S += toadd
        res.append(int(S))
    return res[1:]

```

Les premières valeurs obtenues sont 1, 1, 2, 4, 9, 20, 48, 115, 286, 719.

3. On rappelle que lorsqu'un groupe  $G$  agit sur un ensemble  $E$ , on a pour tout élément  $e \in E$  la formule

$$|\text{Orb}(e)| = \frac{|G|}{|\text{Fix}(e)|}.$$

Considérons l'orbite de  $w$  pour l'action de  $\text{SG}(t)$  sur  $V(t)$ . Si  $x \in \text{Orb}(w)$ , alors tout le sous-arbre  $t_w$  constitué des descendants de  $w$  dans  $t$  est isomorphe au sous-arbre  $t_x$ , car un élément de  $\text{SG}(t)$  préserve les relations de descendance. Il est clair qu'alors, on a aussi

$$t_w \cup \{z \text{ relié à } w\} \simeq_{(\text{isomorphe})} t_x \cup \{z \text{ relié à } x\},$$

donc rajouter une feuille au niveau de tout sommet  $x \in \text{Orb}(w)$  donne un arbre isomorphe à  $t'$ . Réciproquement, si lorsqu'on rajoute une feuille au niveau de  $x$  on obtient un arbre  $t''$  isomorphe à  $t'$ , alors la restriction d'un isomorphisme d'arbres enracinés  $\psi : V(t') \rightarrow V(t'')$  à  $V(t)$  est un élément de  $\text{SG}(t)$  qui envoie  $w$  sur  $x$ . On conclut que

$$\begin{aligned} |\text{Orb}(w)| &= \text{nombre de sommets } x \in V(t) \text{ tels que l'adjonction d'une feuille en } x \text{ donne } t' \\ &= u(t, t'). \end{aligned}$$

Ceci donne la première formule. Pour la seconde, si  $y \in \text{Orb}(z)$  pour l'action de  $\text{SG}(t')$  sur  $V(t')$ , alors les deux arbres  $t = (t')_{\downarrow z}$  et  $(t')_{\downarrow y}$  obtenus en retirant  $z$  ou  $y$  à  $t'$  sont isomorphes, car un élément  $\sigma \in \text{SG}(t')$  envoyant  $z$  sur  $y$  préserve les relations de descendance. Réciproquement, si  $t$  et  $(t')_{\downarrow y}$  sont isomorphes, alors  $y$  est dans l'orbite de  $z$  pour l'action de  $\text{SG}(t')$  : on étend un isomorphisme d'arbres enracinés  $\psi : V(t) \rightarrow V((t')_{\downarrow y})$  en posant  $\psi(z) = y$ . Ainsi,

$$\begin{aligned} |\text{Orb}(z)| &= \text{nombre de sommets } y \in V(t') \text{ tels que la suppression de } y \text{ donne } t' \\ &= d(t, t'). \end{aligned}$$

Finalement, si  $\sigma \in \text{Fix}(w, t)$ , il s'étend en  $\sigma \in \text{Fix}(z, t')$  en posant  $\sigma(z) = z$ , et la réciproque de cette opération est la restriction d'une application de  $V(t')$  à  $V(t)$  (si  $\sigma \in \text{SG}(t')$  fixe  $z$ , alors il fixe aussi son père  $w$ ). On conclut que

$$\frac{|\text{SG}(t)|}{u(t, t')} = |\text{Fix}(w, t)| = |\text{Fix}(z, t')| = \frac{|\text{SG}(t')|}{d(t, t')}.$$

4. En appliquant la formule de la question précédente aux arbres  $s = \bullet$  et  $t \in \mathfrak{T}(N)$ , on obtient :

$$d(t) = |\text{SG}(\bullet)| d(\bullet, t) = |\text{SG}(t)| u(\bullet, t) = |\text{SG}(t)| u(t),$$

donc  $d(t) u(t) = \frac{(d(t))^2}{|\text{SG}(t)|}$  et les deux formules pour  $\mathbb{P}_N[t]$  sont équivalentes. Avec  $t \in \mathfrak{T}(4)$ , on obtient les valeurs suivantes :

$$\mathbb{P}_4 \left[ \begin{array}{c} \circ \\ | \\ \bullet \\ | \\ \bullet \end{array} \right] = \frac{1}{18} \quad ; \quad \mathbb{P}_4 \left[ \begin{array}{c} \circ \\ | \\ \bullet \\ / \quad \backslash \\ \bullet \quad \bullet \end{array} \right] = \frac{1}{9} \quad ; \quad \mathbb{P}_4 \left[ \begin{array}{c} \circ \\ / \quad \backslash \\ \bullet \quad \bullet \\ | \\ \bullet \end{array} \right] = \frac{1}{2} \quad ; \quad \mathbb{P}_4 \left[ \begin{array}{c} \circ \\ / \quad \backslash \\ \bullet \quad \bullet \\ / \quad \backslash \\ \bullet \quad \bullet \end{array} \right] = \frac{1}{3}.$$

On vérifie bien que  $\frac{1}{18} + \frac{1}{9} + \frac{1}{2} + \frac{1}{3} = 1$ .

5. Étant donnés deux arbres  $t$  et  $t'$  de tailles  $N$  et  $N + 1$ , on

$$\langle t | \mathcal{D}_N(t') \rangle = d(t, t') \langle t | t \rangle = |\text{SG}(t)| d(t, t')$$

et

$$\langle \mathcal{U}_N(t) | t' \rangle = u(t, t') \langle t' | t' \rangle = |\text{SG}(t')| u(t, t').$$

Les deux produits scalaires sont égaux par l'identité (1).

6. Avec les notations proposées dans l'énoncé, on a

$$\mathcal{U}_N(t) = \sum_{i=1}^N t^{\uparrow i} \quad ; \quad \mathcal{D}_{N-1}(t) = \sum_{j=r+1}^N t_{\downarrow j}.$$

En effet, par exemple pour  $\mathcal{U}_N(t) = \sum_{t' | t \nearrow t'} u(t, t') t'$ , les facteurs  $u(t, t')$  apparaissent lorsqu'on réunit les arbres  $t^{\uparrow i}$  qui sont isomorphes; et le même argument s'applique pour le calcul de  $\mathcal{D}_{N-1}(t)$ . Notons par ailleurs que les arbres  $t_i$  ou  $t^{\uparrow i}$  ont une numérotation naturelle pour leurs sommets :

- L'arbre  $t^{\uparrow i}$  a pour sommets  $\{v_1, \dots, v_{N+1}\}$ , où  $v_{N+1}$  est le sommet rajouté; les feuilles de  $t^{\uparrow i}$  sont  $\{v_{r+1}, \dots, v_{N+1}\}$  si  $i \leq r$ , et  $\{v_{r+1}, \dots, v_{i-1}, v_{i+1}, \dots, v_{N+1}\}$  si  $i > r$ .
- L'arbre  $t_{\downarrow j}$  a pour sommets  $\{v_1, \dots, v_{j-1}, v_{j+1}, \dots, v_N\}$ .

Alors, avec des notations évidentes :

$$\begin{aligned} \mathcal{D}_N \mathcal{U}_N(t) &= \sum_{i=1}^N \mathcal{D}_N(t^{\uparrow i}) = \sum_{i=1}^r \sum_{j=r+1}^{N+1} (t^{\uparrow i})_{\downarrow j} + \sum_{i=r+1}^N \sum_{\substack{r+1 \leq j \leq N+1 \\ j \neq i}} (t^{\uparrow i})_{\downarrow j} \\ &= N t + \sum_{\substack{1 \leq i \leq N, r+1 \leq j \leq N \\ j \neq i}} (t^{\uparrow i})_{\downarrow j} \end{aligned}$$

car  $(t^{\uparrow i})_{N+1} = t$  pour tout indice  $i$ . Par ailleurs,

$$\mathcal{U}_{N-1} \mathcal{D}_{N-1}(t) = \sum_{j=r+1}^N \mathcal{U}_{N-1}(t_{\downarrow j}) = \sum_{\substack{1 \leq i \leq N, r+1 \leq j \leq N \\ j \neq i}} (t_{\downarrow j})^{\uparrow i}.$$

Le résultat s'en déduit car, si  $i \geq 1$ ,  $j \geq r+1$  et  $j \neq i$ , l'opération de supprimer la feuille  $v_j$  et de rajouter une feuille fille de  $v_i$  commutent, donc  $(t^{\uparrow i})_{\downarrow j} = (t_{\downarrow j})^{\uparrow i}$ .

7. On commence par calculer les coefficients  $c_{w,a,b}(P)$  lorsque le mot est de la forme  $\mathcal{D}\mathcal{U}^a$ . Les premiers termes s'écrivent comme suit :

$$\begin{aligned} \mathcal{D}\mathcal{U} &= \mathcal{U}\mathcal{D} + P \\ \mathcal{D}\mathcal{U}^2 &= \mathcal{U}\mathcal{D}\mathcal{U} + (P+1)\mathcal{U} = \mathcal{U}^2\mathcal{D} + (P+(P+1))\mathcal{U} \\ \mathcal{D}\mathcal{U}^3 &= \mathcal{U}^2\mathcal{D}\mathcal{U} + ((P+1)+(P+2))\mathcal{U}^2 = \mathcal{U}^3\mathcal{D} + (P+(P+1)+(P+2))\mathcal{U}^2 \end{aligned}$$

où sur chaque ligne l'identité est valable dans l'ensemble des applications linéaires de  $\mathbb{R}[\mathfrak{T}(P)]$  dans  $\mathbb{R}[\mathfrak{T}(P+a-1)]$  ( $\mathcal{U}^a$  augmente la taille des arbres de  $a$  et  $\mathcal{D}$  la diminue de 1). On montre par récurrence que

$$(\mathcal{D}\mathcal{U}^a)_{|\mathbb{R}[\mathfrak{T}(P)]} = (\mathcal{U}^a\mathcal{D})_{|\mathbb{R}[\mathfrak{T}(P)]} + (P+(P+1)+\dots+(P+a-1))(\mathcal{U}^{a-1})_{|\mathbb{R}[\mathfrak{T}(P)]}.$$

Supposons le résultat vrai jusqu'au rang  $a$ . Alors,

$$\begin{aligned} (\mathcal{D}\mathcal{U}^{a+1})_{|\mathbb{R}[\mathfrak{T}(P)]} &= (\mathcal{D}\mathcal{U}^a)_{|\mathbb{R}[\mathfrak{T}(P+1)]} \circ \mathcal{U}_{|\mathbb{R}[\mathfrak{T}(P)]} \\ &= (\mathcal{U}^a\mathcal{D} + ((P+1)+\dots+(P+a))\mathcal{U}^{a-1})_{|\mathbb{R}[\mathfrak{T}(P+1)]} \circ \mathcal{U}_{|\mathbb{R}[\mathfrak{T}(P)]} \\ &= (\mathcal{U}^{a+1}\mathcal{D})_{|\mathbb{R}[\mathfrak{T}(P)]} + P(\mathcal{U}^a)_{|\mathbb{R}[\mathfrak{T}(P)]} + ((P+1)+\dots+(P+a))(\mathcal{U}^{a-1})_{|\mathbb{R}[\mathfrak{T}(P)]} \\ &= (\mathcal{U}^{a+1}\mathcal{D})_{|\mathbb{R}[\mathfrak{T}(P)]} + (P+(P+1)+\dots+(P+a))(\mathcal{U}^{a-1})_{|\mathbb{R}[\mathfrak{T}(P)]}. \end{aligned}$$

Démontrons maintenant la relation de récurrence pour les coefficients  $c_{w,a,b}(P)$ . On applique l'opérateur  $\mathcal{D}$  aux deux termes de l'expansion de  $w$  sur  $\mathbb{R}[\mathfrak{T}(P)]$  :

$$\begin{aligned} \sum_{a,b \geq 0} c_{\mathcal{D}w,a,b}(P) \mathcal{U}^a \mathcal{D}^b &= \mathcal{D}w = \sum_{a,b \geq 0} c_{w,a,b}(P) \mathcal{D}\mathcal{U}^a \mathcal{D}^b \\ &= \sum_{a,b \geq 0} c_{w,a,b}(P) (\mathcal{U}^a \mathcal{D} + ((P-b)+(P-b+1)+\dots+(P-b+a-1))\mathcal{U}^{a-1}) \circ \mathcal{D}^b \\ &= \sum_{a,b \geq 0} c_{w,a,b}(P) \mathcal{U}^a \mathcal{D}^{b+1} + c_{w,a,b}(P) ((P-b)+(P-b+1)+\dots+(P-b+a-1))\mathcal{U}^{a-1} \mathcal{D}^b \\ &= \sum_{a,b \geq 0} c_{w,a,b-1}(P) \mathcal{U}^a \mathcal{D}^b + \sum_{a,b \geq 0} c_{w,a+1,b}(P) ((P-b)+(P-b+1)+\dots+(P-b+a))\mathcal{U}^a \mathcal{D}^b \end{aligned}$$

en faisant des changements d'indices à la dernière ligne. Le résultat s'en déduit en identifiant les coefficients.

8. Pour tout arbre  $t' \in \mathfrak{T}(N)$ , on a

$$\sum_{t|t \nearrow t'} d(t) d(t, t') = \sum_{\bullet=t_1 \nearrow t_2 \nearrow \dots \nearrow t_{N-1}=t \nearrow t'} d(t_1, t_2) d(t_2, t_3) \dots, d(t_{N-1}, t') = d(t')$$

donc  $Q_{N,d}$  est une matrice stochastique. La relation (5) est une conséquence immédiate du calcul de la question précédente pour  $(\mathcal{D}\mathcal{U}^a)_{|\mathbb{R}\mathfrak{T}(P)}$ , avec  $a = N - 1$  et  $P = 1$  :

$$(\mathcal{D}\mathcal{U}^{N-1})(\bullet) = (\mathcal{U}^{N-1}\mathcal{D} + (1 + 2 + \dots + N - 1)\mathcal{U}^{N-2})(\bullet) = \binom{N}{2}(\mathcal{U}^{N-2})(\bullet),$$

puisque  $\mathcal{D}(\bullet) = 0$ . Alors, à partir de la formule donnée pour  $S(t)$ , on obtient :

$$\begin{aligned} S(t) &= \binom{N}{2} \frac{1}{|\text{SG}(t)|} \langle t | \mathcal{U}^{N-2}(\bullet) \rangle = \binom{N}{2} \frac{1}{|\text{SG}(t)|} \langle \mathcal{D}^{N-2}(t) | \bullet \rangle \\ &= \binom{N}{2} \frac{d(t)}{|\text{SG}(t)|} = \binom{N}{2} u(t), \end{aligned}$$

la dernière identité ayant été prouvée au début de la question 4.

9. On commence par montrer les formules de transition entre les mesures de Plancherel  $\mathbb{P}_N$  et  $\mathbb{P}_{N-1}$ . On a :

$$\begin{aligned} (\mathbb{P}_N Q_{N,d})(t) &= \sum_{t' \in \mathfrak{T}(N)} \mathbb{P}_N[t'] Q_{N,d}(t', t) = \sum_{t' \in \mathfrak{T}(N)} \frac{u(t') d(t')}{\prod_{i=2}^N \binom{i}{2}} \frac{d(t) d(t, t')}{d(t')} \\ &= \frac{d(t) u(t)}{\prod_{i=2}^{N-1} \binom{i}{2}} \sum_{t' \in \mathfrak{T}(N)} \frac{d(t, t') u(t')}{\binom{N}{2} u(t)} = \mathbb{P}_{N-1}[t] \sum_{t' \in \mathfrak{T}(N)} Q_{N,u}(t, t') = \mathbb{P}_{N-1}[t] \end{aligned}$$

en utilisant pour la dernière identité le caractère stochastique de  $Q_{N,u}$ . De même,

$$\begin{aligned} (\mathbb{P}_{N-1} Q_{N,u})(t') &= \sum_{t \in \mathfrak{T}(N-1)} \mathbb{P}_{N-1}[t] Q_{N,u}(t, t') = \sum_{t \in \mathfrak{T}(N-1)} \frac{u(t) d(t)}{\prod_{i=2}^{N-1} \binom{i}{2}} \frac{d(t, t') u(t')}{\binom{N}{2} u(t)} \\ &= \frac{d(t') u(t')}{\prod_{i=2}^N \binom{i}{2}} \sum_{t \in \mathfrak{T}(N-1)} \frac{d(t) d(t, t')}{d(t')} = \mathbb{P}_N[t'] \sum_{t \in \mathfrak{T}(N-1)} Q_{N,d}(t', t) = \mathbb{P}_N[t'] \end{aligned}$$

puisque  $Q_{N,d}$  est une matrice stochastique. Ceci implique que  $\mathbb{P}_N$  est invariante par  $Q_N$  :

$$\mathbb{P}_N Q_N = \mathbb{P}_N Q_{N,d} Q_{N,u} = \mathbb{P}_{N-1} Q_{N,u} = \mathbb{P}_N.$$

En fait, la mesure est même réversible, car

$$\begin{aligned} \mathbb{P}_N[s] Q_N(s, t) &= \sum_{s \searrow r \nearrow t} \mathbb{P}_N[s] Q_{N,d}(s, r) Q_{N,u}(r, t) \\ &= \sum_{s \searrow r \nearrow t} \frac{1}{\binom{N}{2} \prod_{i=2}^N \binom{i}{2}} \frac{d(r) d(r, s) d(r, t) u(s) u(t)}{u(r)} \end{aligned}$$

est une expression symétrique en  $s$  et  $t$ , donc égale à  $\mathbb{P}_N[t] Q_N(t, s)$ . Il reste à montrer le caractère irréductible et apériodique. L'apériodicité est claire, car  $Q_N(t, t) > 0$  pour tout  $t$  : on peut rester en  $t$  lors d'une transition en supprimant une feuille  $w$  (par  $Q_{N,d}$ ), puis en la rajoutant au même endroit (par  $Q_{N,u}$ ). Pour l'irréductibilité, on peut atteindre de tout arbre  $t \in \mathfrak{T}(N)$  l'arbre spécial  $s$  avec une racine et  $N - 1$  feuilles directement en dessous de la racine : il suffit de couper une à une les feuilles et de les regreffer au niveau de la racine. On a donc  $P^k(t, s) > 0$  pour tout  $t$  et  $k$  assez grand, et comme la chaîne est réversible, ceci implique aussi  $P^k(s, t) > 0$  pour  $k$  assez grand.

Le théorème fournit un algorithme pour construire un arbre aléatoire  $\mathfrak{T}(N)$  suivant *exactement* la loi  $\mathbb{P}_N$  (on n'utilise pas le théorème ergodique). En effet, on a

$$\mathbb{P}_N = \mathbb{P}_1 Q_{2,u} Q_{3,u} \cdots Q_{N,u},$$

donc si l'on part de l'arbre  $\bullet$  et que l'on choisit aléatoirement des feuilles à greffer suivant les mesures de transition  $Q_{i,u}$ ,  $2 \leq i \leq N$ , alors on obtient au bout de  $N - 1$  greffes un arbre suivant la mesure de Plancherel  $\mathbb{P}_N$ .

10. On pouvait, par exemple :

- construire les matrices de transitions  $Q_{N,d}$  et  $Q_{N,u}$ , disons avec  $N = 5$  ou  $6$ , et vérifier qu'elles sont stochastiques et qu'elles entrelacent les mesures de Plancherel pour deux valeurs de  $N$ .
- programmer la chaîne de Markov décrite dans la troisième partie du texte, pour une valeur  $N$  petite par exemple égale à  $N = 5$  ou  $6$ .
- écrire un programme qui, à partir d'un mot en les opérateurs  $\mathcal{U}$  et  $\mathcal{D}$ , calcule sa forme normale au sens de l'équation (4) du texte.

Ici, on va être un peu ambitieux et implanter dans Python un programme qui tire au hasard les arbres enracinés sous une mesure de Plancherel  $\mathbb{P}_N$ . La difficulté principale est la manipulation des arbres enracinés; on utilisera librement certains packages de Sage/Python, et il était aussi possible de recourir à d'autres modules Python indépendants. Un arbre enraciné s'obtient avec la classe `RootedTree` : par exemple, `A = RootedTree([[[]], [], []])` est l'arbre de l'énoncé de la question 1. On calcule la taille d'un arbre et son nombre d'élagage avec les fonctions suivantes :

```
def size(A):
    return A.node_number()

def d(A):
    N = taille(A)
    return math.factorial(N)/prod([x.node_number() for x in A.subtrees()])
```

Appliqué à l'arbre précédent, on obtient `size(A) = 5` et `d(A) = 12`. Pour les facteurs de symétries, notons que si  $t$  est un arbre enraciné dont les fils de la racine portent des sous-arbres  $t_1, \dots, t_r$ , alors pour engendrer une symétrie  $\sigma \in \text{SG}(t)$ , on peut :

- choisir des symétries  $\sigma_1, \dots, \sigma_r$  de chacun des sous-arbres  $t_1, \dots, t_r$ ;
- puis, permuer parmi les  $t_i$  les sous-arbres qui sont isomorphes.

Le facteur de symétrie  $|\text{SG}(t)|$  vérifie donc la relation de récurrence :

$$|\text{SG}(t)| = \left( \prod_{i=1}^r |\text{SG}(t_i)| \right) |\text{Sym}(t_1, \dots, t_r)|,$$

où le second facteur dans le terme de droite est le produit des factorielles des multiplicités des types d'arbres enracinés qui apparaissent comme sous-arbre basé en l'un des fils de la racine. Le programme suivant implémente ce calcul :

```
def SG(A):
    N = size(A)
    if N == 1:
```

```

    return 1
else:
    child_trees = list(A.subtrees())[1:A.node_number_at_depth(1)+1]
    res = prod([SG(x) for x in child_trees])
    d = {}
    for x in child_trees:
        if x in d.keys():
            d[x] += 1
        else:
            d[x] = 1
    res *= prod([math.factorial(d[x]) for x in d.keys()])
    return res

```

On obtient avec l'arbre de taille 5 précédemment considéré  $SG(A) = 2$ . Les fonctions  $u(t)$  et  $\mathbb{P}_N[t]$  sont ensuite obtenues par les fonctions

```

def u(A):
    return d(A)/SG(A)

```

```

def Plancherel(A):
    N = size(A)
    if N == 1:
        return 1
    else:
        num = (d(A))^2
        denom = SG(A) * prod([binomial(i,2) for i in range(2,N+1)])
        return num/denom

```

qui donnent pour notre exemple  $u(A) = 6$  et  $Plancherel(A) = 2/5$ . On vérifie le calcul de la question 4 en affichant la liste `RootedTrees(4)` :

```
[[[[[]]], [[[], []]], [[], [[]]], [[], [], []]]
```

et leurs probabilités avec `[Plancherel(A) for A in RootedTrees(4)]` :

```
[1/18, 1/9, 1/2, 1/3]
```

Pour les probabilités de transition  $Q_{N,u}$  et  $Q_{N,d}$ , le plus simple est de calculer les coefficients  $u(t, t')$  en utilisant le fait que  $\sum_{i=1}^N t^{\uparrow i} = \sum_{t \nearrow t'} u(t, t') t'$ , voir question 6. Les arbres  $t^{\uparrow i}$  sont obtenus en rajoutant une feuille au niveau du sommet étiqueté  $i$ , pour un certain étiquetage des sommets. Il faut donc jongler entre arbres enracinés étiquetés (`LabelledRootedTree`) et non étiquetés (`RootedTree`) :

```

def graft_list_with_multiplicities(A):
    LA = LabelledRootedTree(A)
    L = map(lambda LB : RootedTree(LB), A.graft_list(LabelledRootedTree([])))
    res = {}
    for B in L:
        if B in res.keys():
            res[B] += 1
        else:
            res[B] = 1

```

```

return res

def uu(A,B):
    if B in A.graft_list(RootedTree([])):
        return graft_list_with_multiplicities(A)[B]
    else:
        return 0

def dd(A,B):
    if B in A.graft_list(RootedTree([])):
        return graft_list_with_multiplicities(A)[B] * SG(B) / SG(A)
    else:
        return 0

```

Avec A comme précédemment et  $B = \text{RootedTree}([\text{[]}], [], [], [\text{[]}])$ , on obtient  $uu(A,B) = 1$  (on peut obtenir B à partir de A qu'en rajoutant une feuille à la racine), et  $dd(A,B) = 3$  (on peut obtenir A à partir de B en supprimant l'une des trois feuilles de profondeur 1). Finalement, pour simuler un arbre enraciné de taille  $N$  sous la mesure de Plancherel  $\mathbb{P}_N$ , on utilise une fonction qui rajoute aléatoirement une feuille à  $t$  en suivant la mesure de transition  $Q_{|t|+1,d}$ :

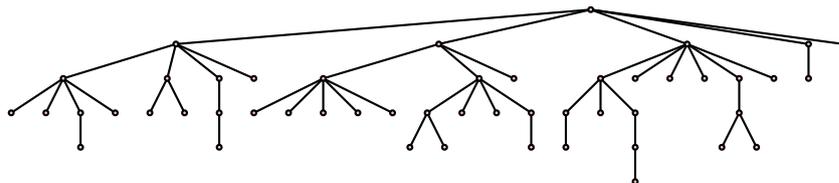
```

def add_random_leaf(A):
    d = graft_list_with_multiplicities(A)
    L = d.keys()
    N = 1+size(A)
    denum = binomial(N,2)*u(A)*SG(A)
    prob_transition = []
    for i in range(0,len(L)):
        B = L[i]
        prob_transition.append( uu(A,B) * u(B) * SG(B) / denum )
    return L[GeneralDiscreteDistribution(prob_transition).get_random_element()]

def PlancherelRandom(N):
    count = 1
    res = RootedTree([])
    while count < N:
        res = add_random_leaf(res)
        count += 1
    return res

```

Par exemple, un arbre aléatoire de taille 50 sous la mesure de Plancherel est dessiné ci-dessous :



Les propriétés asymptotiques de tels arbres aléatoires (profondeur, degré moyen des sommets, forme, *etc.*) ne semblent pas connues à ce jour.