Dénombrement d’arbres enracinés

Année scolaire 2019/2020

Noé Chirot, Mathéo Izac,Tom Laporte, Robin Mabilat, Jean Malvezin, Harilala Rahaerindraibe, Taddéo Rocq, Thomas Sanchez, Inès Termentina et Alice Vidal, élèves de terminale S.

Encadré par Alexandre Rocq

Établissement : Lycée Émile Duclaux, Aurillac

Chercheur : Dominique Manchon, chargé de recherches au CNRS, Laboratoire de Mathématiques Blaise Pascal, CNRS et Université Clermont- Auvergne

I) Présentation du sujet

Notre sujet est de compter le nombre d'arbres enracinés différents pouvant exister avec un nombre de sommets donné.

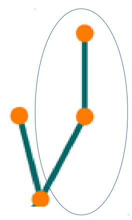
Un **arbre enraciné,** pour donner dans un premier temps une définition simple, est en fait, graphiquement, une succession de sommets et d'arêtes se construisant de bas en haut, sachant que de chaque sommet peut partir n'importe quel nombre d'arêtes. Les arêtes ne se croisent jamais et ne se rejoignent jamais par le haut, et à leur deux extrémités se trouvent un sommet.

Les **arêtes** sont symbolisés par des traits simples, des segments :

Les **sommets** liant les arêtes sont représentés par des points :

Une **branche** est un ensemble d'arêtes et de points directement issu de la division de l'arbre au niveau du premier sommet.

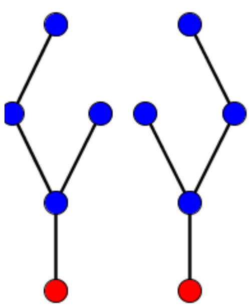
Par exemple, ici se trouve un arbre possédant 4 sommets et 3 arêtes :



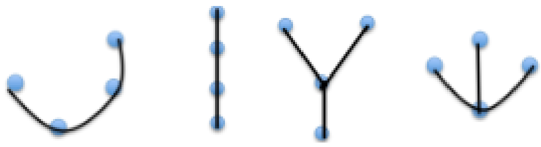
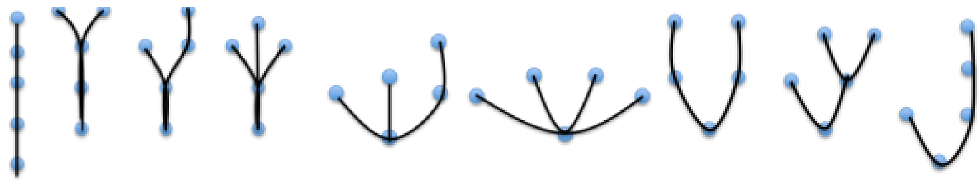
Une branche

Pour compter les arbres différents, nous avons du établir certaines règles pour ne pas compter deux fois le même arbre : deux arbres sont **identiques** si on obtient le deuxième à l'aide d'une suite d'opérations consistant à choisir un sommet v du premier arbre, et à permuter les branches du sous-arbre issu de v.

Exemple de deux arbres identiques :



Quelques exemples :

Arbres à 1 sommets	
Arbres à 2 sommets	
Arbres à 3 sommets	
Arbres à 4 sommets	
Arbres à 5 sommets	

Les arbres de degré inférieur ou égal à 5 restent relativement faciles à trouver, mais cela devient bien plus complexe à partir de 6 sommets sans « classer » ces arbres, ce qui nous a amené à essayer de trouver un classement, défini à partir des arbres avec un nombre moindre de sommets, et donc une formule de récurrence nous permettant de calculer le nombre d'arbres en fonction du nombre de sommets, ce qui est en fait l'objet de cet article.

II) Conjectures et résultats obtenus

Voici quelques résultats importants démontrés au cours de l'article :

1) Les combinaisons avec répétitions, c'est-à-dire le nombre de façons de choisir k éléments parmi n éléments, en ayant la possibilité de choisir plusieurs fois le même élément parmi les n éléments pour constituer l' ensemble à k éléments, noté $\begin{bmatrix} n \\ k \end{bmatrix}$ est égal à $\begin{bmatrix} n \\ k-1 \end{bmatrix} + \begin{bmatrix} n-1 \\ k \end{bmatrix}$.

2) Le lien entre combinaisons avec répétitions et combinaisons sans répétition est $\begin{bmatrix} n \\ k \end{bmatrix} = \begin{pmatrix} n+k-1 \\ k \end{pmatrix}$

Remarque : Dans la dernière partie de l'article, nous présentons une formule générale permettant de calculer le nombre d'arbres différents en fonction du nombre de sommets, mais présenter cette formule dès à présent serait complexe et sans doute malvenu, dans la mesure ou elle est assez longue et fait appel à une nomenclature assez conséquente introduite tout au long de l'article, et incompréhensible hors contexte.

III) Le dénombrement d'arbres enracinés

1) Une définition plus « formelle » d'un arbre enraciné

Un **arbre enraciné** est un graphe représenté sur un plan. Il est orienté vers le haut : il possède un seul sommet racine à partir duquel partent k arêtes (k pouvant être nul). Chaque arête possède un sommet à chacune de ses extrémités. Des sommets de ces premières arêtes issues de la racine peuvent à nouveau partir d'autres arêtes, s'il part de nouvelles arrêtes de ceux-ci, il sont donc de nouvelles racines, créant ainsi des sous-arbres.

Remarque : les arêtes ne se croisent jamais et ne se rejoignent pas par le haut.

La **racine** ou **sommet racine** est le sommet à l'origine de l'arbre, chaque arbre ne peut posséder qu'une seule racine.

Une **branche** est un ensemble d'arêtes directement issu de la division de l'arbre à la racine.

Le **degré n** de l'arbre est défini par le nombre de sommets qu'il possède.

Le **nombre u_n** est le nombre d'arbres différents qu'il est possible de créer avec n sommets.

Le **rang k** indique le nombre d'arêtes partant de la racine, c'est aussi le nombre de branches.

Le **niveau** d'un sommet est en fait son étage, le niveau 0 est celui de la racine, le niveau 1 est celui où se trouvent les k sommets issus de la racine.

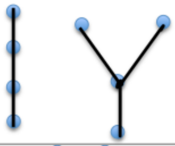
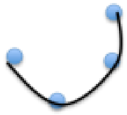
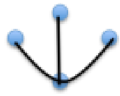
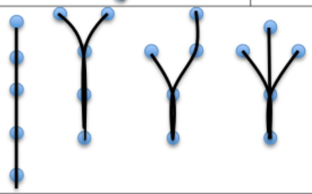
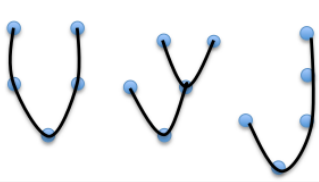
Pour rappel, deux arbres sont **identiques** si on obtient le deuxième à l'aide d'une suite d'opérations consistant à choisir un sommet v du premier arbre, et à permuter les branches du sous-arbre issu de v.

2) Recherche à la main.

A) Quelques résultats…

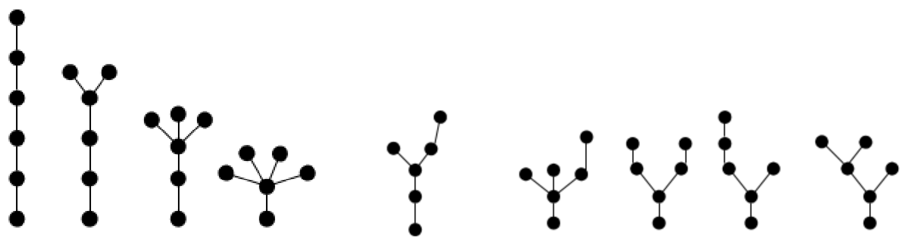
Degré	n=1	n=2	n=3	n=4	n=5	n=6
Nombre d'arbres différents :	$u_1=1$	$u_2=1$	$u_3=2$	$u_4=4$	$u_5=9$	$u_6=20$

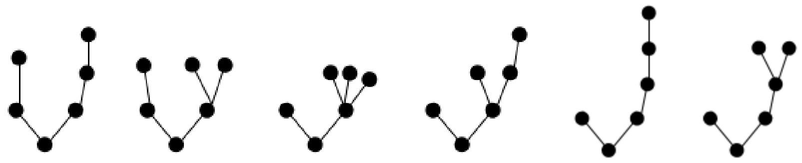
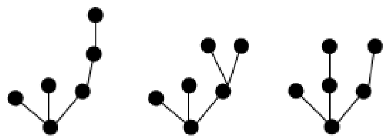
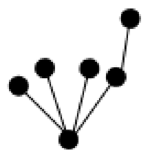
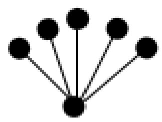
B) … triés par rangs.

K / N	1	2	3	4
1				
2				
3				
4				
5				

Remarque : Les arbres de degré inférieur ou égal à 5 restent relativement faciles à trouver, même sans trier par rang, mais cela devient bien plus complexe à partir de n=6.

Pour n=6 :

k=1	

k=2	
k=3	
k=4	
k=5	

3) Sommes de produits et partitions.

Pour compter le nombre d'arbres que l'on peut faire avec n sommets, il nous est apparu qu'il fallait le faire par récurrence, en se servant des résultats pour des arbres de degré moindre.

A) Somme :

Pour faire cela, nous avons remarqué qu'il était possible de compter tous les arbres différents en faisant varier k. En effet il s'agit de sommer les configurations possibles pour k=1 puis k=2 puis k=3 et ainsi de suite jusqu'à k = n - 1 dans un arbre à n sommets (comme il y a n sommets au total, dont un qui est le sommet racine, $1 \leqslant k \leqslant n-1$)

B) Partition :

Il faut donc trouver une méthode pour répartir les n-1 sommets restants sur les k arêtes partant de la racine. Pour cela, il faut utiliser la partition des entiers : trouver toutes les possibilités pour répartir n-1 sommets sur k arêtes revient à trouver toutes les partitions de l'entier n-1 en somme de k entiers strictement positifs.

On définit donc **P(a, b)** comme l'ensemble des partitions de **a** en exactement **b** entiers, avec $1 \leq b \leq a$. Une partition de l'ensemble s'écrit alors $\{\mathbf{r_1, r_2, r_3, r_4, \dots, r_{a-b+1}}\}$ avec :

- $r_1, r_2, r_3, r_4, \dots, r_{a-b+1}$ des entiers naturels.
- $a = 1 \times r_1 + 2 \times r_2 + 3 \times r_3 + 4 \times r_4 + \dots + (a-b+1)\, r_{a-b+1}$
- $b = r_1 + r_2 + r_3 + r_4 + \dots + r_{a-b+1}$

Remarque : r_i est donc le nombre de fois qu'apparaît l'entier i dans la partition. Pour revenir à notre problème, c'est le nombre de branches qui vont recevoir i sommets.

Remarque : on ne peut pas utiliser des entiers supérieurs à a-b+1 dans une partition car si tel était le cas, il y aurait alors moins de b entiers dans cette partition.

Exemple : ensemble des partitions de l'entier 5 en 3, notés P(5, 3)
5 = 1 + 1 + 3 = 1 x 2 + 2 x 0 + 3 x 1 d'où la partition {2, 0, 1}
5 = 1 + 2 + 2 = 1 x 1 + 2 x 2 + 3 x 0 d'où la partition {1, 2, 0}

d'où **P(5, 3) = { {2, 0, 1}, {1, 2, 0} }**

Maintenant que nous avons une méthode pour trouver la partition de chaque entier défini en fonction du nombre d'arêtes issues de la racine, pour connaître le nombre d'arbres différents en fonction de k, il faut sommer le nombre d'arbres différents de chaque partition.

C) Produit :

Néanmoins, il reste à calculer le nombre d'arbres différents pour chaque partition donnée.
Nous avons d'abord pensé qu'il s'agissait du produit du nombres de configurations de chaque branche.

1/ Exemple où il n'y a que deux branches :

Prenons un exemple où n=8 et k=2. Prenons la partition pour laquelle la première branche a quatre sommets et la deuxième en a trois.
En isolant la première branche (B1), lorsqu'on part du sommet de niveau 1, il y a $u_4=4$ configurations différentes. Sur la seconde branche (B2), en partant du sommet de niveau 1, il y a $u_3=3$ configurations différentes.
On peut commencer par fixer B1 dans une configuration donnée appelée C1.
Faisons varier B2. Elle varie 2 fois. Deux arbres différents sont créés.
Fixons B1 dans une autre configuration C2.
Faisons varier B2. Elle varie à nouveau 2 fois. Deux arbres différents sont créés.
Le même processus est réitéré jusqu'à ce que B1 ait pris toutes les configurations possibles, pendant que B2 varie 2 fois pour chaque configuration différente de B1.
Ainsi, comme B1 varie 4 fois, il y a $4 \times 2 = u_4 \times u_3$ arbres différents créés avec cette partition.
Il est évident que sur deux branches seulement, cela fonctionne de la même manière, sauf s'il y a le même nombre de sommets distribués sur plusieurs branches, mais nous verrons cette exception dans la partie suivante.

2/ Exemple où il n'y a que trois branches, il s'agit aussi d'un produit :

Soit un arbre de degré n où k=3 : il y a trois branches B1, B2 et B3 de tailles distinctes.
Prenons B1 avec p configurations différentes, B2 avec q configurations différentes, et B3 avec r configurations différentes
Etape 1 : on peut fixer B1 et B2, faire varier B3 autant de fois qu'elle le peut. Ensuite on change B2 de configuration une seule fois, si cela est possible, et on fait varier la troisième à nouveau. On réitère le processus jusqu'à ce que la deuxième branche se soit retrouvé dans toutes les configurations possibles.
B1 n'a pas varié, B2 a varié q fois et B3 r fois. Il y a donc eu rq arbres créés.

Etape 2 : faisons varier B1 une fois, si cela est possible. Réitérons le processus décrit en étape 1. Il y a donc eu rq arbres créés à nouveau.
En refaisant ce processus jusqu'à ce que B1 se soit retrouvé dans toutes les configurations possibles, c'est-à-dire p fois, il y a eu pqr arbres créés . Il s'agit donc bien du produit du nombre de configurations de chaque branche.

3/ Généralisation : Démontrons cette propriété par récurrence :

Soit P_k la propriété : « Le nombre d'arbres créés (noté A_k) pour un entier naturel k supérieur ou égal à 2 donné et une partition *P* donnée est égale au produit du nombre de configurations qu'il existe sur chaque branche (noté u_i , i variant de 1 à k, comme il y a k branches), sauf dans le cas particulier où au moins deux branches ont le même nombre de sommets. »

Initialisation: faite ci-dessus dans l'**exemple 1/.**
Hérédité: Supposons $\exists k \in \mathbb{N} \setminus \{0;1\} / P_k \textit{ soit vraie}$ Montrons que P_{k+1} l'est aussi.

Par hypothèse de récurrence, il existe un arbre de rang k de partition P, dont toutes les branches ont un nombre de sommets différents, donc un nombre différent de configurations différentes u_i , et pour cette partition, $A_k=\prod_{i=1}^{k} u_i$.

Au rang k+1, on ajoute une branche, dont le nombre de sommets est différent des k autres.
Pour cette branche, il existe u_{k+1} configurations différentes. Si on fixe cette branche sur une configuration donnée, en faisant varier les autres branches, on a par hypothèse, A_k arbres différents. En changeant de configuration la branche rajoutée et en refaisant varier les autres, encore A_k nouveaux arbres sont créés. Procédons de même jusqu'à ce que la dernière branche se soit trouvée dans toutes les configurations différentes possibles.

Cette branche a varié u_{k+1} fois, donc $A_{k+1}=u_{k+1}\times A_k=u_{k+1}\times\prod_{i=1}^{k} u_i=\prod_{i=1}^{k+1} u_i$

P_{k+1} est donc vraie. La propriété est héréditaire.

<u>Conclusion :</u> Par application du principe de récurrence, P_k est vraie pour tout entier naturel k strictement positif. $\forall n\in\mathbb{N}\setminus\{0;1\}, A_k=\prod_{i=1}^{k} u_i$.

Il est donc désormais établi que pour un rang k donné et une partition donnée, le nombre d'arbres pour cette partition est égal au produit du nombre de configurations qu'il existe sur chacune des k branches, sauf dans le cas où il y a des branches avec le même nombre de sommets, cas que nous allons éclaircir dès maintenant.

<u>4) Constatation de l'existence de « doublons » :</u>

<u>*A) Rappels préalables :*</u>

1/ n factoriel, noté « n ! », est le produit de tout les entiers naturels inférieurs ou égaux à n, c'est-à-dire : $n!=1\times2\times3\times...\times(n-1)\times n=\prod_{k=1}^{n} k$

2/ Rappels sur les coefficients binomiaux :

a) Définition :les **coefficients binomiaux, ou combinaisons sans répétition**, définis pour tout entier naturel n et tout entier naturel k inférieur ou égal à n donnent le nombre d'ensembles à k éléments dans un ensemble de n éléments.

b) Notation : Ils sont notés $\binom{n}{k}$.

c) La formule explicite pour calculer des combinaison sans répétitions est : $\binom{n}{k}=\frac{n!}{k!(n-k)!}$

d) La relation du triangle de Pascal est : $\binom{n}{k}=\binom{n-1}{k}+\binom{n-1}{k-1}$ (elle est ici admise, mais se démontre assez facilement par un raisonnement ensembliste)

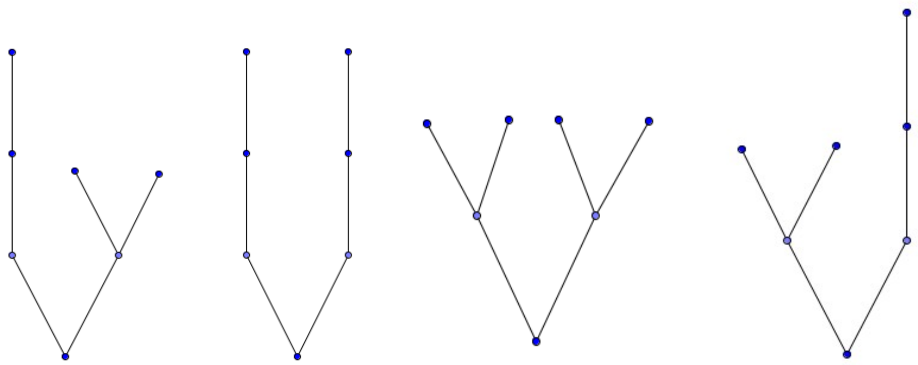
B) Que sont les « doublons » ?

Lorsqu’on se retrouve dans une configuration où il y a, à l’issue d’une division,
k sommets du même niveau recevant le même nombre de sommets chacun (au delà de 2), il y a la possibilité qu’apparaissent deux arbres identiques dits « doublons » en utilisant la technique du produit du nombre de configurations différentes de chaque branche expliquée dans la partie précédente.

Prenons l’exemple le plus simple possible : n=7 et k=2 et chacun des 2 sommets reçoit 2 sommets en plus. On a alors ce nombre d’arbres avec le produit : $U_3 . U_3 = 2 . 2 = 4$.
Mais en réalité on obtient quatre arbres dont deux, nous le verrons, sont identiques :

- Prenons A1, fixons la branche de gauche
- Faisons varier la branche de droites, A2 est formé.
- Changeons de configuration la branche de gauche, on obtient un nouvel arbre : A3.
- Ensuite, changeons la branche de droite de configuration. A4 est formé.

Mais on constate que l’arbre A1 et l’arbre A4 sont identiques : il n’y a pas quatre arbres différents comme l’indiquait le simple produit mais seulement 3 arbres différents, il existe ce que nous avons appelé un « doublon ».



A1 A2 A3 A4

Remarque 1 : Il ne peut exister des arbres identiques que si au moins deux sommets du même niveau reçoivent le même nombre de sommets.
Si, par exemple, le premier sommet reçoit p sommets et le second reçoit q sommets (avec p différent de q), il est évident que la première branche et la deuxième branche ne peuvent être identiques, elles ne sont pas du même degré.
Remarque 2 : Le nombre de sommets reçu en plus par k sommets issus de la racine doit être supérieur ou égal à 2 pour qu’il y ait des « doublons ». Si le nombre de sommets reçu en plus est de 1 sur k sommets du même niveau, les branches sont toutes identiques : il n’y a qu’un arbre qui peut être formé avec deux sommets, il n’y a donc pas d’arbres identiques en faisant un simple produit : 1x1x1x...x1=1.

C) Compter sans doublons

On cherche donc à trouver une méthode différente que le simple produit pour dénombrer les arbres, ou du moins, une façon d'éliminer les doublons dans une configuration ou k sommets du même niveau reçoivent le même nombre de sommets en plus (supérieur ou égal à 2).

Remarque 1 : Pour des soucis de simplification de nomenclature, nous avons choisi que le nombre de sommets ajouté créerait n configurations différentes possibles sur chaque branche. Ce « n » n'a donc rien à voir avec le n indiquant le degré de l'arbre, en réalité, il faudrait noter u_n si on ajoutait n-1 sommets.

Remarque 2 : Le nombre d'arbres différents sans doublons lorsque n branches différentes sont possibles sur k sommets du même niveau est noté $\begin{bmatrix} n \\ k \end{bmatrix}$, c'est le nombre de manières de choisir k éléments parmi n avec répétitions possibles.

Pour dénombrer les arbres sans doublons dans un tel cas, nous avons regardé des cas particuliers afin d'être en capacité d'émettre une hypothèse.

1/ Lorsque 2 sommets reçoivent le même nombre de sommets en plus

Dans ce cas il y a donc 2 sommets de même niveau qui sont racines de n arbres différents.

Le nombre d'arbres différents est donc ici $\begin{bmatrix} n \\ 2 \end{bmatrix}$.

Il existe n configurations pour chacune des branches.
Fixons la branche de gauche (appelée branche 1 ou B1), par exemple, dans une configuration donnée (appelée configuration 1 ou C1), faisons varier l'autre branche (branche 2 ou B2) dans toutes les configurations possibles, n arbres différents se sont formés.
Fixons à présent la B1 dans une configuration différente (configuration 2 ou C2). En faisant à nouveau varier la B2 dans toutes les configurations, on constate la présence d'un doublon : un arbre identique a nécessairement déjà été formé auparavant. En effet l'arbre pour lequel la B1 se trouvait dans la C1 et la B2 dans la C2 (cet arbre a existé nécessairement car la B2 a connu toutes les configurations lorsque que la B1 était en C1) apparaît ici mais c'est la B1 qui est en C2 et la B2 qui est en C1. Ainsi seulement n-1 nouveaux arbres différents ont été formés.
En fixant la B1 en C3, et en faisant varier la B2 dans toutes les configurations possibles, on constate la présence de deux doublons : celui formé quand B1 était en C1 et B2 en C3, et celui quand B1 était en C2 et B2 en C3. Il y a donc n-2 nouveaux sommets formés.
En continuant ainsi, on s'aperçoit qu'à la fixation de la configuration i de B1, il y a i-1 doublons formé en plus, soit n-(i-1) nouveaux arbres différents. Cela se démontre assez facilement par récurrence sur i.
Ainsi on constate que le nombre d'arbres différents formés au total est :

$$\begin{bmatrix} n \\ 2 \end{bmatrix} = n + n - 1 + n - 2 + \ldots + n - (n-1) + n - n = \sum_{k=1}^{n} k = \frac{n(n+1)}{2}$$

2/ Lorsque 3 sommets reçoivent le même nombre de sommets

Dans ce cas il y a donc 3 sommets de même niveau qui sont racines de n arbres différents.

Le nombre d'arbres différents est donc ici $\begin{bmatrix} n \\ 3 \end{bmatrix}$.

Pour dénombrer les arbres différents, procédons par disjonction de cas.
Soit les trois branches sont identiques, soit toutes différentes, soit il y en a deux d'identiques, et une différente.
- Dans le cas où elles sont toutes identiques, il y a n arbres différents, car il existe n configurations différentes sur chaque branche, et on veut que les trois branches soient identiques.
- Dans le cas où les trois branches sont différentes, il y a $\binom{n}{3}$ arbres différents, car cela revient à choisir toutes les combinaisons de 3 éléments différents parmi n sans que deux branches soient identiques.
- Dans le cas où il y a seulement deux branches identiques, on peut procéder en regardant d'abord uniquement les deux branches identiques : il existe n arbres différents, puisque ces deux branches sont identiques et qu'il y a n configurations possibles par branches. Cependant il reste la troisième branche, qui elle aussi varie n fois, il faut donc multiplier n par n. Néanmoins en faisant cela, nous n'avons pas éliminé les cas où les trois branches sont identiques, elles le sont n fois, on retire donc n à n^2. Ainsi il y a $n^2-n=n(n-1)$ nouveaux arbres différents formés.

Donc, somme toute,

$$\begin{bmatrix} n \\ 3 \end{bmatrix}=n+\binom{n}{3}+n(n-1)=n+\frac{n!}{3!\times(n-3)!}+n(n-1)=n+\frac{n(n-1)(n-2)}{6}+n(n-1)$$

En mettant au même dénominateur, et en factorisant par n on a :

$$\begin{bmatrix} n \\ 3 \end{bmatrix}=\frac{6n+n(n-1)(n-2)+6n(n-1)}{6}=\frac{n(6+(n-1)(n-2)+6(n-1))}{6}=\frac{n(6+n\times n-3n+2+6n-6)}{6}$$

En factorisant de nouveau, on obtient :

$$\begin{bmatrix} n \\ 3 \end{bmatrix}=\frac{n(n+1)(n+2)}{6}$$

Compter le nombre d'arbres différents lorsque n configurations différentes sont possibles sur k sommets, sachant que des branches identiques issues de plusieurs des k sommets sont admises pour créer un arbre, revient en fait à compter le nombre d'ensembles à k éléments (ici les k branches) parmi n éléments (ici, les n configurations pour chaque branche) avec répétition (la même configuration peut-être choisie sur plusieurs branches).
Ne connaissant aucune formule ni explicite ni de récurrence concernant les combinaisons avec répétitions, nous avons cherché à lier les combinaisons avec répétitions et sans répétitions, c'est-à-dire avec des coefficients binomiaux.
Tout d'abord, pour k=1, il n'y a qu'une branche sur laquelle se trouve n configurations. Ainsi il y a n arbres différents.

Donc $\begin{bmatrix} n \\ 1 \end{bmatrix}=n=\frac{n!}{1!(n-1)!}=\binom{n}{1}=\binom{n+1-1}{1}$

Reprenons les résultats pour k=2 et k=3

$$\begin{bmatrix} n \\ 2 \end{bmatrix}=\frac{n(n+1)}{2}=\frac{(n+1)!}{2!(n-1)!}=\binom{n+1}{2}=\binom{n+2-1}{2}$$

$$\begin{bmatrix} n \\ 3 \end{bmatrix} = \frac{n(n+1)(n+2)}{6} = \frac{(n+2)!}{3!(n-1)!} = \binom{n+2}{3} = \binom{n+3-1}{3}$$

D'après les résultats obtenus, on peut conjecturer que le nombre de façons de choisir k éléments parmi n avec répétitions est égal au nombre de façons de choisir k éléments parmi n+k-1 sans répétitions.

Hypothèse : $\begin{bmatrix} n \\ k \end{bmatrix} = \binom{n+k-1}{k}$.

Nous allons donc dans la prochaine partie démontrer cela par récurrence.

D) Démonstration :

1/ Partie 1 de la démonstration de l'hypothèse de récurrence faite en III)c/

Soit un ensemble à n éléments distincts. On cherche à déterminer le nombre de façons de choisir un ensemble à k éléments avec répétitions parmi les n éléments de l'ensemble, « avec répétitions » signifie qu'un élément peut être choisi plusieurs fois.

Ce nombre est noté $\begin{bmatrix} n \\ k \end{bmatrix}$ lu « k parmi n avec répétitions ».

N'ayant aucune formule explicite pour calculer ce nombre à ce point de l'article, il faut établir une relation triangulaire (de récurrence) en se servant des termes précédents, comme avec la relation du triangle de Pascal sur les combinaisons sans répétition. Nous avons alors trouvé la formule suivante :

Formule : $\begin{bmatrix} n \\ k+1 \end{bmatrix} = \begin{bmatrix} n \\ k \end{bmatrix} + \begin{bmatrix} n-1 \\ k+1 \end{bmatrix}$ ou autrement dit : $\begin{bmatrix} n \\ k \end{bmatrix} = \begin{bmatrix} n \\ k-1 \end{bmatrix} + \begin{bmatrix} n-1 \\ k \end{bmatrix}$

Démonstration : Procédons de façon ensembliste. Je considère mon ensemble à n éléments distincts, et cherche donc à déterminer le nombre de façons de choisir un ensemble à k éléments avec répétitions parmi les n éléments de l'ensemble.

Pour cela, procédons par disjonction de cas : fixons un élément de l'ensemble à n éléments.

Soit cet élément est dans mon ensemble à k éléments, et donc, il me reste k-1 éléments à choisir avec répétitions pour compléter l'ensemble à k éléments parmi n éléments restants (n toujours, car l'élément fixé peut être choisi d'autres fois), dans ce cas on note $\begin{bmatrix} n \\ k-1 \end{bmatrix}$; soit l'élément fixé n'est pas dans dans l'ensemble à k éléments, il reste donc k éléments à choisir avec répétitions parmi n-1 éléments (puisque l'élément fixé n'y est pas), on note cela $\begin{bmatrix} n-1 \\ k \end{bmatrix}$.

Le « soit…, soit » signifie une somme en logique. Donc $\begin{bmatrix} n \\ k \end{bmatrix} = \begin{bmatrix} n \\ k-1 \end{bmatrix} + \begin{bmatrix} n-1 \\ k \end{bmatrix}$

2/ Partie 2 de la démonstration de l'hypothèse de récurrence faite en III)c)

La relation triangulaire désormais établie, il est possible de réaliser une récurrence sur k, pour prouver notre hypothèse de récurrence liant combinaison avec et sans répétitions.

Hypothèse : Soit P_k : « $\begin{bmatrix} n \\ k \end{bmatrix} = \binom{n+k-1}{k} \forall n \in \mathbb{N} \setminus \{0\}, \forall k \in \mathbb{N} \setminus \{0\}$ »

Initialisation : P_0 : « $\begin{bmatrix} n \\ 1 \end{bmatrix} = \begin{pmatrix} n \\ 1 \end{pmatrix} \forall n \in \mathbb{N} \setminus \{0\}$ »

En créant un ensemble à 1 élément parmi n éléments, la question de la répétition ne se pose pas, le nombres de façons de choisir 1 élément parmi n est toujours n, avec ou sans répétitions.

Donc $\begin{bmatrix} n \\ 1 \end{bmatrix} = \begin{pmatrix} n \\ 1 \end{pmatrix} \forall n \in \mathbb{N} \setminus \{0\}$. P_0 est vraie, la propriété est initialisée.

Hérédité : Supposons $\exists p \in \mathbb{N} \setminus \{0\} / \forall k \in [1 ; p], \begin{bmatrix} n \\ k \end{bmatrix} = \begin{pmatrix} n+k-1 \\ k \end{pmatrix}$ Montrons que la propriété est vraie jusqu'au rang p+1 (c'est-à-dire $\begin{bmatrix} n \\ p+1 \end{bmatrix} = \begin{pmatrix} n+p \\ p+1 \end{pmatrix} \forall n \in \mathbb{N} \setminus \{0\}$)

On a d'après la relation triangulaire trouvée ci-devant :

$$\begin{bmatrix} n \\ p+1 \end{bmatrix} = \begin{bmatrix} n-1 \\ p+1 \end{bmatrix} + \begin{bmatrix} n \\ p \end{bmatrix} = \begin{bmatrix} n-2 \\ p+1 \end{bmatrix} + \begin{bmatrix} n-1 \\ p \end{bmatrix} + \begin{bmatrix} n \\ p \end{bmatrix} = \begin{bmatrix} n-3 \\ p+1 \end{bmatrix} + \begin{bmatrix} n-2 \\ p \end{bmatrix} + \begin{bmatrix} n-1 \\ p \end{bmatrix} + \begin{bmatrix} n \\ p \end{bmatrix} = \dots\dots$$

On peut ainsi décomposer le terme de gauche jusqu'à $\begin{bmatrix} 1 \\ p+1 \end{bmatrix}$, ce qui dans notre sujet correspond à disposer 1 sommet de plus ou, du moins, une seule configuration possible sur chacun des p+1 sommets. On obtient donc un seul arbre.

En décomposant jusqu'à là, on obtient finalement :

$$\begin{bmatrix} n \\ p+1 \end{bmatrix} = \begin{bmatrix} 1 \\ p+1 \end{bmatrix} + \sum_{i=2}^{n} \begin{bmatrix} i \\ p \end{bmatrix} = 1 + \sum_{i=2}^{n} \begin{pmatrix} i+p-1 \\ p \end{pmatrix} (\textit{on a supposé la propriété vraie pour tout entier naturel } n)$$

Ainsi en changeant d'indice, on a :

$$\begin{bmatrix} n \\ p+1 \end{bmatrix} = 1 + \sum_{i=p+1}^{n+p-1} \begin{pmatrix} i \\ p \end{pmatrix}$$

Or la relation du triangle de Pascal dit que : $\begin{pmatrix} i+1 \\ p+1 \end{pmatrix} = \begin{pmatrix} i \\ p+1 \end{pmatrix} + \begin{pmatrix} i \\ p \end{pmatrix} \Leftrightarrow \begin{pmatrix} i+1 \\ p+1 \end{pmatrix} - \begin{pmatrix} i \\ p+1 \end{pmatrix} = \begin{pmatrix} i \\ p \end{pmatrix}$

Ainsi $\begin{bmatrix} n \\ p+1 \end{bmatrix} = 1 + \sum_{i=p+1}^{n+p-1} \begin{pmatrix} i \\ p \end{pmatrix} = 1 + \sum_{i=p+1}^{n+p-1} \left(\begin{pmatrix} i+1 \\ p+1 \end{pmatrix} - \begin{pmatrix} i \\ p+1 \end{pmatrix} \right) (\textit{il s'agit d'une somme téléscopique})$

D'où $\begin{bmatrix} n \\ p+1 \end{bmatrix} = 1 + \sum_{i=p+1}^{n+p-1} \left(\begin{pmatrix} i+1 \\ p+1 \end{pmatrix} - \begin{pmatrix} i \\ p+1 \end{pmatrix} \right) = 1 + \begin{pmatrix} n+p \\ p+1 \end{pmatrix} - \begin{pmatrix} p+1 \\ p+1 \end{pmatrix} = 1 + \begin{pmatrix} n+p \\ p+1 \end{pmatrix} - 1 = \begin{pmatrix} n+p \\ p+1 \end{pmatrix}$

P_{p+1} est vraie, la propriété est héréditaire.

Conclusion : Par application du principe de récurrence, P_k est vraie pour tout entier naturel k strictement positif.

$$\begin{bmatrix} n \\ k \end{bmatrix} = \begin{pmatrix} n+k-1 \\ k \end{pmatrix} \forall n \in \mathbb{N} \setminus \{0\}, \forall k \in \mathbb{N} \setminus \{0\}$$

5) Formule finale :

A) La formule et son explication

La suite (u_n) qui compte le nombre d'arbres de degré n est alors définie par :

$$u_1 = 1$$

$$u_{n+1} = \sum_{k=1}^{n} \left(\sum_{p=1}^{Card\ P(n,k)} \left(\prod_{i=1}^{n-k+1} \left[\begin{matrix} u_i \\ r_{i(P_p)} \end{matrix} \right] \right) \right)$$

Explication de la formule :

- La première somme additionne successivement tous les arbres possibles de rang 1, puis de rang 2, jusqu'au rang n (un arbre de rang n+1 ne pouvant exister).
  La variable k représente donc le rang de l'arbre considéré.

- Pour chaque rang k, il reste donc n sommets à distribuer sur k arêtes, d'où l'ensemble P(n, k) dont chaque élément est une partition qui représente la manière de distribuer les n sommets. La deuxième somme va donc additionner toutes les combinaisons possibles pour chaque partition dans l'ensemble P(n, k).
  La variable p correspond au numéro de la partition considérée.
  Exemple: Pour une partition $\mathbf{\{r_1, r_2, r_3, r_4, \dots, r_{n-k+1}\}}$ de l'ensemble P(n, k), on compte donc tous les arbres de degré n+1 et de rang k, dont r_1 arêtes qui partent de la racine reçoivent 1 sommet supplémentaire, r_2 arêtes qui partent de la racine reçoivent 2 sommets supplémentaires, … et r_{n-k+1} arrêtes partant de la racine reçoivent n-k+1 sommets supplémentaires.

- Pour expliquer le dernier terme, servons-nous d'un exemple. On a dit que chaque arête reçoit un certain nombre de sommets : on peut dire aussi que chaque arête reçoit un arbre d'un certain degré. Si on considère un arbre de rang 2, dont la première arête reçoit un arbre de rang 4 et la deuxième arête un arbre de rang 5, il y aura donc u_5 x u_6 arbres différents de rang 2. Sauf que si deux arêtes reçoivent le même nombre de sommets, ce simple produit ne suffit plus (problème déjà évoqué plus haut) et il faut donc utiliser les combinaisons avec répétitions. D'où leur utilisation pour les groupes d'arêtes qui reçoivent le même nombre de sommets (avec $r_{i\ (P\ p)}$ qui correspond au nombre d'arêtes qui reçoivent un arbre de rang i, u_i correspond au nombre d'arbres de degré i) et l'utilisation du produit pour les autres cas.

B) Application de la formule générale au calcul de u_7 et u_8.

Pour passer de $u_6=20$ à $u_7=48$ sans tous les dessiner, on compte les arbres par rang, du rang 1 jusqu'au rang 6 :

- il y a **$u_6=20$** arbres de degré 7 et de rang 1

- Pour trouver tous les arbres de rang 2, on trouve toutes les partitions de 6 en deux entiers :
  6 = 5 + 1
  6 = 4 + 2
  6 = 3 + 3
  Il y a donc $u_5 \times u_1 + u_4 \times u_2 + \begin{bmatrix} u_3 \\ 2 \end{bmatrix} = 9 \times 1 + 4 \times 1 + 3 = 16$ arbres de rang 2
  Remarque : on rappelle que si on avait fait u_3 x u_3, on aurait compter un arbre de trop.

- Pour trouver tous les arbres de rang 3, on trouve toutes les partitions de 6 en trois entiers :
  6 = 4 + 1 + 1
  6 = 3 + 2 + 1
  6 = 2 + 2 + 2
  Il y a donc $u_4 \times u_1 \times u_1 + u_3 \times u_2 \times u_1 + u_2^3 = 4 + 2 + 1 = 7$ arbres de rang 3

- Pour trouver tous les arbres de rang 4, on trouve toutes les partitions de 6 en quatre entiers :
  6 = 3 + 1 + 1 + 1
  6 = 2 + 2 + 1 + 1
  Il y a donc $u_3 \times u_1^3 + u_2^2 \times u_1^2 = 2 + 1 = 3$ arbres de rang 4

- Pour trouver tous les arbres de rang 5, on trouve toutes les partitions de 6 en cinq entiers :
  6 = 2 + 1 + 1 + 1 + 1
  Il y a donc $u_2 \times u_1^4 = 1$ arbre de rang 5

- Il y en plus un arbre de rang 6

d'où $u_7 = 20 + 16 + 7 + 3 + 1 + 1 = 48$

Avec le même raisonnement pour u_8, on trouve :

$$u_8 = u_7 + u_6 + u_5 \times u_2 + u_4 \times u_3 + u_5 + u_4 \times u_2 + \begin{bmatrix} u_3 \\ 2 \end{bmatrix} + u_3 \times u_2^2 + u_4 + u_3 \times u_2 + u_2^2 + u_3 + u_2^2 + u_2 + u_1^6$$
$$u_8 = 48 + 20 + 9 + 8 + 9 + 4 + 3 + 2 + 4 + 2 + 1 + 2 + 1 + 1 + 1 = 115$$

Par souci de clarté, les multiplications par u_1 (u_1 étant égal à 1) n'ont pas été écrites.

C) Un programme d'application de la formule générale à u_n :

Le programme suivant (écrit en python) est une application de la formule expliquée dans le premier paragraphe de la partie 5.

```
from math import * # pour factorial(n)


# Dans ce programme, une partition {r(1), r(2), r(3), r(4), ..., r(a-b+1)} est
représentée par un tableau à 3 dimensions :
# 1 ère  -> ensemble de partitions
# 2 ième -> nombre d'occurrences r(i) de l'entier i
# 3 ième -> entier i
# Cette fonction calcule toutes les différentes partitions possibles et prend
comme arguments :
# a = entier à partitioner
# b = nombre de partitions
# c = entier maximal à utiliser
# Cette fonction trouve les partitions par récurrence informatique
def partitions(a, b, c) :
	# cas particuliers et initialisation
	if a==0 :
		return [[[0, 0]]]
	elif b>=a or a==1 :
		return [[[a, 1]]]
	elif b==1 :
		return [[[1, a]]]

	l=[] #liste où seront stockées les partitions trouvées

	h=a/b # -> ?
	if h!=int(h) : h=h+1
	h=int(h)

	for i in range(min(a-b+1, c), h-1, -1) : # utilisation de min() car on
ne peut pas avoir une partition avec un entier > a-b+1
		# récurrence informatique
		ens=partitions(a-i, b-1, i)

		for part in ens :
			if part[0][1] == i :
				part[0][0]+=1
			else :
				part=[[1, i]]+part

			l.append(part) # on ajoute la partition trouvée
	return l


# fonction pour afficher les partitions de manière lisible (pas utilisée dans la
suite, c'était juste pour les tests)
def aff_part(prt) :
	for p in prt :
		partition_str=""
		for i in p :
			for nbr in range(0,i[0]) :
				partition_str+=str(i[1])
		print(partition_str)
```

```
# premiers termes de la suite u(n)
un_l = [ 1, 1, 1, 2, 4, 9 ]


# Combinatoire
def C(n, k) :
        if k==1 :
                return n
        elif n==k  :
                return 1
        else :
                return int(factorial(n)/(factorial(k)*factorial(n-k))) #
simplification à faire pour diminuer le nombre de calculs


def u(n) :

        if n<len(un_l) : # si le terme a déjà été calculé
                return un_l[n]

        res = 0 # variable qui va contenir le résultat (initialisation à 0 car
addition)

        for k in range (1, n) : # pour chaque rang

                for part in partitions(n-1, k, n-1) : # pour chaque partition
pour un certain rang

                        inter = 1 # variable intermédiaire (initialisation à 1
car multiplication)

                        for nbr in part : # pour chaque entier de la partition
considérée (nbr[0] -> r(i), nbr[1] -> i)
                                inter *= C(u(nbr[1])+nbr[0]-1, nbr[0])

                        res+=inter

        un_l.append(res) # on ajoute le nouveau terme à notre liste

        return res

#afficher les 50 premiers termes de la suite
for a in range(0,50) :
        print("{}{}{}{}".format("u(", a, ") = ", u(a)))
```

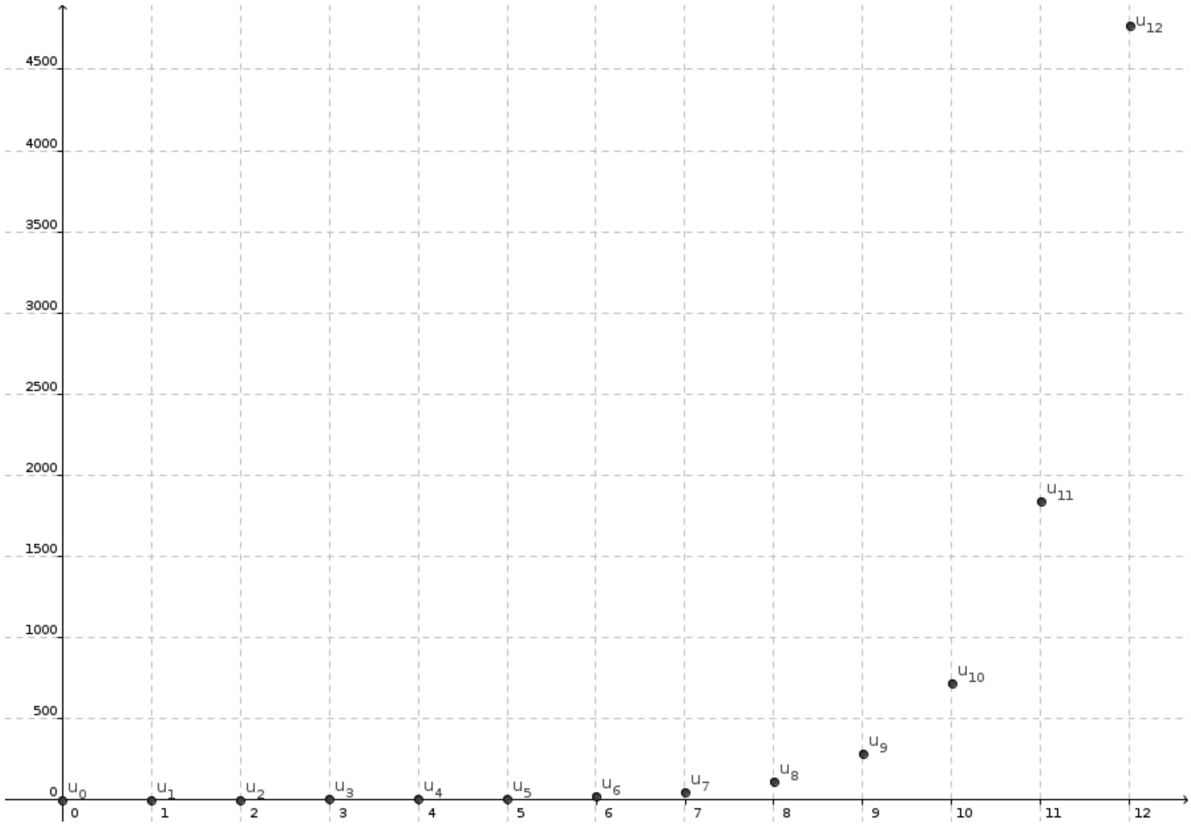
Voici un tableau présentant les premiers résultats de ce programme :

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
u_n	1	1	1	2	4	9	20	48	115	286	719	1842	4766	12486	32973

n	15	16	17	18	22	23	24
u_n	87811	235381	634847	1721159	4688676	12826228	35221832

Remarque : Les résultats donnés par ce programme ont été vérifié sur *The one-line encyclopedia of integer sequences.*

Et enfin voici un graphique des premiers termes de la suite :



Conclusion :

Au final, nous sommes parvenus à établir une formule de récurrence fonctionnelle pour compter le nombre d'arbres enracinés différents suivant un nombre de sommets donné.
Cette formule fait intervenir entre autres choses le principe de partition d'un entier et de combinaisons. Si cette formule est efficiente, elle demeure assez pénible à lire, aussi doit-il être possible de la simplifier, voire de trouver une formule explicite pour compter le nombre d'arbres différents, ce que nous n'avons pas eu le temps de faire.

Pour finir, nous tenons à remercier chaleureusement M. Manchon, le chercheur qui nous a proposé ce sujet, et nous a donné des pistes de recherches lors de plusieurs séminaires, à M. Chaumat qui nous a également aidé lors des séminaires par ses conseils éclairés, ainsi que M. Rocq, le professeur de mathématiques qui nous a accompagné tout au long de l'année, afin que nous puissions finalement écrire cet article.
Et enfin, merci à l'association MATh.en.JEANS .