

Cet article est rédigé par des élèves. Il peut comporter des oublis et imperfections, autant que possible signalés par nos relecteurs dans les notes d'édition.

Un problème de polymères

Année 2023 – 2024

Adèle Grimault, Raphaëlle Magnin, Pierre Even, élèves de terminale

Établissements : Lycée Honoré d'Estienne d'Orves, Carquefou ; Lycée Grand Air, La Baule

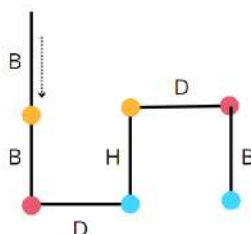
Enseignant-es: Sandrine Bulliard, Mikaël Jousseau, Bastien Issartel, Bertrand Bordonado

Chercheur : Elric Angot, Université de Nantes.

1. Présentation du sujet

On s'intéresse à un problème de biologie. Des polymères sont constitués de chaînes d'atomes pouvant aller vers le haut (H), vers le bas (B) ou à droite (D), elles ne peuvent en aucun cas aller vers la gauche ou retourner sur leurs pas. Combien de chaînes à **1000** atomes existent ?

Exemple :



Ci-dessus une représentation possible d'un polymère composé de six atomes.

2. Résultats

Nous avons adopté trois approches différentes pour résoudre ce problème : une approche basée sur les graphes, une autre sur la somme de suites, et enfin une approche utilisant une unique suite récurrente que nous avons pu expliciter. Grâce à ces approches, nous avons pu obtenir un résultat exact du nombre de possibilités pour une chaîne de 1000 atomes, en utilisant deux programmes Python l'un reprenant la méthode des sommes de suites, l'autre de la suite récurrente.

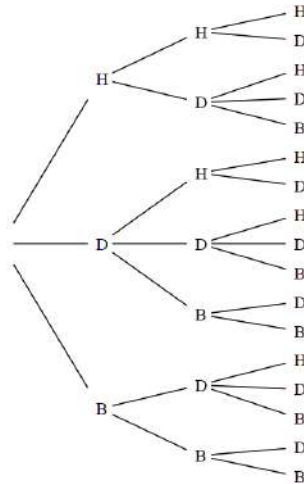
Nombre de possibilités pour un polymère de 1000 atomes :

```
720163369435338750561314684442472393287231976284407517972018980635880883127002019
434829484771095362037402066496127027299201700013544541071734804839626055194931177
898217584577678589862270198056506390025669464968653646665435628263033778777008772
661352762091252780372044430424871530892268495446812453002601671410252771564827375
68934079466850318276966893735585103845471745828701580706481
```

3. Méthodes de résolution

3.1. Graphe et matrice d'adjacence

Tout d'abord, nous avons modélisé la situation par un arbre des possibilités afin de compter les possibilités de chaînes pour un atome, deux atomes...



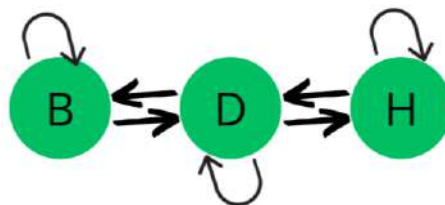
On pose n le nombre le nombre d'atome dans la chaîne. On observe que :

- pour $n = 1$, il y a une possibilité
- pour $n = 2$, il y a trois possibilités
- pour $n = 3$, il y a 7 possibilités
- pour $n = 4$, il y a 17 possibilités

On observe également que :

- quand on est allé à D à n , on a trois possibilités à $n + 1$: aller en haut, aller à droite et aller en bas.
- quand on est allé en H à n , on a deux possibilités à $n + 1$: aller en haut et aller à droite. Ici, on ne peut pas aller en bas car cela reviendrait à retourner sur nos pas.
- quand on est allé à B à n , on a deux possibilités à $n + 1$: aller à droite et aller en bas. Ici, on ne peut pas aller en haut car cela reviendrait à retourner sur nos pas.

On modélise cette situation par le graphe ci-dessous.



On peut ensuite associer ce graphe à sa matrice d'adjacence [\(1\)](#) :

$$\begin{matrix} & \begin{matrix} D & B & H \end{matrix} \\ \begin{matrix} D \\ B \\ H \end{matrix} & \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \end{matrix}$$

Comme une matrice d'adjacence à la puissance n nous donne le nombre de chemins possibles de longueur n partant de X et arrivant à Y (2) et qu'au départ de notre polymère on a trois possibilités :

- aller à droite,
- aller en haut,
- aller en bas,

tout comme après D, c'est comme si notre chaîne partait de D. Donc pour obtenir le nombre de chaînes de n atomes possibles on met notre matrice à la puissance n qui nous intéresse et on additionne les chemins partant de D et arrivant à D, les chemins partant de D et arrivant à H et les chemins partant de D et arrivant à B.

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}^n$$

Exemple pour $n = 2$:

$$\begin{aligned} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}^2 &= \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 \times 1 + 1 \times 1 + 1 \times 1 & 1 \times 1 + 1 \times 1 + 1 \times 0 & 1 \times 1 + 1 \times 0 + 1 \times 1 \\ 1 \times 1 + 1 \times 1 + 0 \times 1 & 1 \times 1 + 1 \times 1 + 0 \times 0 & 1 \times 1 + 1 \times 0 + 0 \times 1 \\ 1 \times 1 + 0 \times 1 + 1 \times 1 & 1 \times 1 + 0 \times 1 + 1 \times 0 & 1 \times 1 + 0 \times 0 + 1 \times 1 \end{pmatrix} \\ &= \begin{pmatrix} 3 & 2 & 2 \\ 2 & 2 & 1 \\ 2 & 1 & 2 \end{pmatrix} \end{aligned}$$

$3 + 2 + 2 = 7$, c'est bien le nombre de possibilités au rang 2 !

La matrice d'adjacence devient cependant difficile à utiliser lorsque les valeurs deviennent trop grandes.

3.2. Somme de suites

Nous avons en parallèle envisagé une autre approche : le nombre de chaînes d'atomes possibles peut être décomposé en la somme des chemins qui mènent en haut, à droite et en bas.

En effet, on a constaté sur l'arbre des possibilités que pour être en haut, il faut venir de H ou de D, pour être à droite il faut venir de H, D ou B et pour être en bas il faut venir de D ou de B.

On a donc pu établir les suites ci-dessous, h_n , d_n et b_n désignant les nombres de chaînes de longueur n se terminant respectivement par H, D et B.

$$\begin{aligned} h_{n+1} &= h_n + d_n & h_{n+1} &= 1h_n + 1d_n + 0b_n \\ d_{n+1} &= h_n + d_n + b_n & d_{n+1} &= 1h_n + 1d_n + 1b_n \\ b_{n+1} &= b_n + d_n & b_{n+1} &= 0h_n + 1d_n + 1b_n \end{aligned}$$

Pour exploiter ces suites interdépendantes, on utilise une suite de matrices.

On pose

$$S_n = \begin{pmatrix} h_n \\ d_n \\ b_n \end{pmatrix} \quad S_0 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

En repérant les coefficients du système de suites (3), on obtient l'expression explicite de notre suite de matrices :

$$\begin{pmatrix} h_n \\ d_n \\ b_n \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}^n \times \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

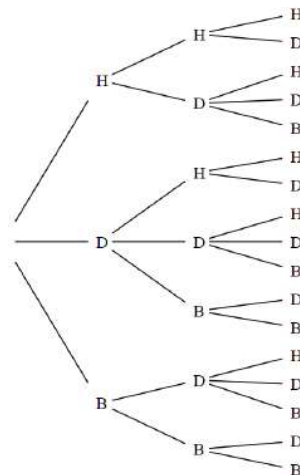
En se basant sur cette méthode nous avons créé un algorithme python, nous permettant d'obtenir le résultat (voir 2- Résultats) :

```
l=int(input("combien d'atomes possède la chaine ?"))
d=3
h=2
b=2
n=0
while n<l-2:
    n=n+1
    D=d+h+b
    H=d+h
    B=d+b
    d=D
    h=H
    b=B
    S=d+h+b
print("Nombre de polymères possibles :", end="") ; print(S)
```

3.3. Une suite récurrente qui mène à une suite explicite

3.3.1. Suite récurrente

En reprenant notre arbre des possibilités on observe :



- Chaque direction prise à l'étape n offre deux nouvelles possibilités de directions à l'étape $n + 1$, sauf si la direction prise est la droite ; dans ce cas, trois possibilités s'offrent. Par conséquent, le nombre de directions à l'étape $n + 1$, que nous appellerons u_{n+1} , est le double du nombre de possibilités à l'étape n (soit $2u_n$) plus une direction supplémentaire pour chaque direction droite à l'étape n .
- À chaque nœud, il y a toujours une unique possibilité d'aller vers la droite. Par conséquent, le nombre de possibilités à l'étape n nous donne le nombre de directions droite à l'étape $n + 1$. À l'inverse le nombre de directions droite à $n + 1$ nous est donné par le nombre de possibilités à n (u_n).

On peut donc exprimer une suite récurrente d'ordre 2 telle que

$$u_{n+2} = 2u_{n+1} + u_n$$

A l'aide de cette suite, nous avons réalisé un programme Python pour calculer le résultat :

```
# Créé par GRIMAULT1, Le 19/12/2023 en Python 3.
from math import*
n=int(input("Entrez du texte : combien d'atomes possède la chaîne ?"))

# 5 premières valeurs des possibilités calculées manuellement
X=[3,7,17,41,99]

# expression algorithmique de la suite  $U_{n+2}=2(U_{n+1})+U_n$ 
for i in range(5,n):
    x=X[i-2]+2*X[i-1]
    X.append(x)
print("le nombre de possibilités est :")
print(X[n-1])
```

3.3.2. Forme explicite

3.3.2.a) Explicitation à l'aide d'une matrice

Tout d'abord, nous avons explicité notre suite sous la forme d'une matrice, pour cela on pose une matrice U telle que

$$U = \begin{pmatrix} u_n \\ u_{n+1} \end{pmatrix}$$

On a donc $U_0 = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$ car $u_0 = 1$ et $u_1 = 3$.

On peut trouver la forme explicite de cette suite de matrices en cherchant la matrice carrée A telle que $U_n = A^n \times U_0$.

Comme $U_{n+1} = A \times U_n$, on pose un système d'équations avec les valeurs connues, et on trouve

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}$$

On a donc

$$U_n = \begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}^n \times \begin{pmatrix} 1 \\ 3 \end{pmatrix}$$

3.3.2.b) Explicitation à l'aide d'une suite

Pour expliciter notre suite récurrente, on va utiliser son équation caractéristique : $x^2 = 2x + 1$, ou $x^2 - 2x - 1 = 0$.

On trouve les racines de ce polynôme :

$$R_1 = 1 - \sqrt{2} \text{ et } R_2 = 1 + \sqrt{2}$$

Or pour une suite récurrente d'ordre 2, on peut trouver une suite explicite de la forme [\(4\)](#)

$$u_n = \alpha \times R_1^n + \beta \times R_2^n$$

Grâce aux valeurs trouvées manuellement de u_1 et u_2 , on trouve un système d'équations :

$$\begin{aligned} 3 &= \alpha \times (1 - \sqrt{2})^1 + \beta \times (1 + \sqrt{2})^1 \\ 7 &= \alpha \times (1 - \sqrt{2})^2 + \beta \times (1 + \sqrt{2})^2 \end{aligned}$$

Par substitution on trouve

$$\alpha = \frac{1 - \sqrt{2}}{2} \text{ et } \beta = \frac{1 + \sqrt{2}}{2}$$

On a donc

$$u_n = \frac{1 - \sqrt{2}}{2} \times (1 - \sqrt{2})^n + \frac{1 + \sqrt{2}}{2} \times (1 + \sqrt{2})^n$$

4. Conclusion

En conclusion, nous avons pu finaliser notre étude en résolvant le problème de différentes manières, grâce à diverses approches basées sur les graphes, les sommes de suites, et les suites récurrentes. Grâce à ces méthodes, nous avons obtenu un résultat exact pour le nombre de possibilités d'une chaîne de 1000 atomes en utilisant un programme Python. Nous tenons à remercier nos professeurs pour leur soutien et leurs conseils tout au long de ce projet.

Notes d'édition

(1) Le coefficient de la ligne X et de la colonne Y de cette matrice est égal à 1 s'il y a une flèche de X vers Y dans le graphe et à 0 sinon (on pourrait aussi inverser le rôle des lignes et des colonnes).

Les lecteurs qui ne connaissent pas le calcul matriciel peuvent aller directement au §3.2 puis sauter l'expression matricielle avant le programme, qui ne l'utilise pas. De même au §3, pour le résultat complet et le second programme, il n'est pas nécessaire de lire le §3.3.2a.

(2) Ceci est un peu expliqué au §3.2 (voir note 3).

(3) S_0 représente le D ajouté au départ comme expliqué plus haut.

La matrice des coefficients des équations de récurrence sur h_n , d_n et b_n est aussi la matrice d'adjacence du graphe (avec des choix différents de ceux du §3.1).

En notant M cette matrice, la relation de récurrence s'écrit $S_{n+1} = MS_n$.

On a ainsi $S_1 = MS_0$, $S_2 = M(MS_0) = M^2S_0$, la multiplication des matrices étant associative, et par récurrence $S_n = M(M^{n-1}S_0) = M^nS_0$ pour tout $n \geq 1$.

Pour interpréter les coefficients de M^n : comme le seul coefficient non nul de S_0 est $d_0 = 1$, le produit $S_n = M^nS_0$ est égal à la colonne centrale de M^n ; ses coefficients sont h_n , d_n et b_n , les nombres de chaînes de longueur n depuis D vers H, D et B. De même la première colonne comporte la répartition des chaînes partant de H et la dernière celle des chaînes partant de B.

(4) L'équation caractéristique permet de déterminer les suites géométriques (r^n) vérifiant la relation de récurrence : on voit simplement que cela revient à dire que r est une solution de l'équation caractéristique, et ici ce sont les deux suites (R_1^n) et (R_2^n) .

Les suites $(\alpha R_1^n + \beta R_2^n)$ où α et β sont des constantes vérifient encore la relation de récurrence. En calculant α et β de façon que $u_1 = \alpha R_1 + \beta R_2$ et $u_2 = \alpha R_1^2 + \beta R_2^2$, on obtient une suite $(\alpha R_1^n + \beta R_2^n)$ satisfaisant la même relation de récurrence d'ordre 2 que (u_n) et ayant les mêmes deux premiers termes. Alors, il en résulte par récurrence que $u_n = \alpha R_1^n + \beta R_2^n$ pour tout $n \geq 1$ (à noter que l'on aurait pu partir de u_0 et u_1).