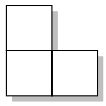
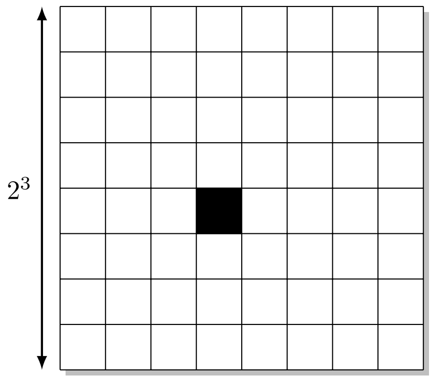
TD1 : récurrence

1 Pavage

Le but de cet exercice est de prouver qu'une grille de taille $2^n \times 2^n$ peut être remplie par des triominos en L en laissant un trou unique, quelle que soit la position du trou :



un triomino en L.



1.1. Pour vous convaincre que le pavage est possible, commencez par prouver que la grille contient le bon nombre de cases. Par récurrence, montrez que pour tout $n \geq 0$, $2^n \times 2^n - 1$ est un multiple de 3.

1.2. Vérifiez ensuite le résultat pour toute position du trou dans une grille de taille 4×4.

1.3. Montrez le résultat par récurrence, en utilisant le fait qu'une grille de côté 2^{n+1} peut être découpée en quatre grilles de taille 2^n.

2 Fibonacci

La suite de Fibonacci présente de nombreuses propriétés intéressantes. On la retrouve dans de nombreux problèmes mathématiques ou algorithmiques. Elle est définie de la manière suivante :

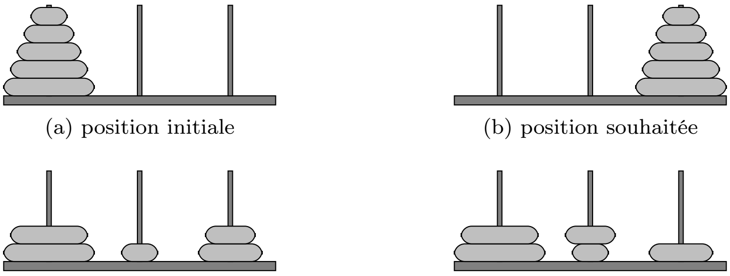
$$\begin{cases} f_0 = 1 \\ f_1 = 1 \\ f_i = f_{i-1} + f_{i-2} \text{ pour tout } i \geq 2 \end{cases}$$

Ses premiers termes sont : 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, ...

2.1. On vous a probablement déjà maintes fois proposé de donner un algorithme pour calculer la suite de Fibonnacci, et découragé de proposer la version récursive ci-dessous.

```
Fonction Fibonacci() → entier naturel
    données : n : entier naturel
    Algorithme
        si n = 0 ou n = 1 alors
            retourner 1
        sinon
            retourner Fibonacci(n − 1)+Fibonacci(n − 2)
```

Pouvez vous dire en fonction de n combien d'additions seront réalisées ? On demande un ordre de grandeur : de l'ordre de n ? n^2 ? a^n ? $!n$?



(a) position initiale (b) position souhaitée

(c) position valide (d) position invalide

Figure 1 – Tours de Hanoï

2.2. Donnez une version itérative permettant de calculer le i$^{\text{ème}}$ terme d'une suite de Fibonacci.

Quelle est cet fois l'ordre de grandeur du nombre d'additions ? En complexité asymptotique la complexité d'un programme est mesurée en fonction de la taille de l'entrée. Si le nombre n est l'entrée, quelle est en réalité l'espace mémoire en terme de nombre de bits pour le stocker ? Quelle serait donc la complexité de votre algorithme en fonction de ce nombre b de bits ?

3 Tours de Hanoï

Le problème des tours de Hanoï consiste à déplacer une tour pyramidale constituée de disques de diamètres différents (les plus larges dessous) d'un emplacement de départ à un emplacement d'arrivée. On dispose pour cela d'un emplacement intermédiaire et on se doit de respecter les règles suivantes :

— on ne peut déplacer plus d'un disque à la fois ;

— on ne peut pas placer un disque sur un autre disque plus petit que lui.

Ces règles sont illustrées sur la Figure 1.

> N. Claus de Siam a vu, (...) dans le grand temple de Bénarès, au-dessous du dôme qui marque le centre du monde, trois aiguilles de diamant, plantées dans une dalle d'airain, hautes d'une coudée et grosses comme le corps d'une abeille. Sur une de ses aiguilles, Dieu enfila au commencement des siècles, 64 disques d'or pur, le plus large reposant sur l'airain, et les autres, de plus en plus étroits, superposés jusqu'au sommet. C'est la tour sacrée du Brahmâ. Nuit et jour, les prêtres se succèdent sur les marches de l'autel, occupés à transporter la tour de la première aiguille sur la troisième, sans s'écarter des régles fixes que nous venons d'indiquer, et qui ont été imposées par Brahma. Quand tout sera fini, la tour et les brahmes tomberont, et ce sera la fin des mondes !
>
> Edouard Lucas, Récréations mathématiques

3.1. Écrivez une procédure récursive permettant de déplacer les disques pour passer de la position de départ à la position d'arrivée. Vous supposerez que vous disposez d'une fonction Déplacer(départ, arrivée) pour déplacer un disque de la colonne « départ » vers la colonne « arrivée ».

Procédure Hanoi

données :

— n : nombre de disques à déplacer

— départ : la colonne de départ

— arrivée : la colonne d'arrivée

— intermédiaire : la colonne restante

précondition : *Une pyramide d'au moins n disques positionnée sur « départ », les autres emplacements ne comportant que des pyramides de sommets plus larges que les n disques les plus hauts de « départ ».*

postcondition : *La pyramide constituée des n disques les plus hauts de « départ » a été déplacée sur celle de l'emplacement « arrivée ».*

Algorithme

...

3.2. Il existe une version itérative (n'utilisant pas une pile) au problème des tours de Hanoï. Saurez vous la trouver ? [1]

1. Indice : examinez le comportement du petit disque.