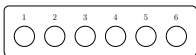
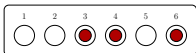
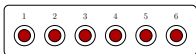


Vider la ligne

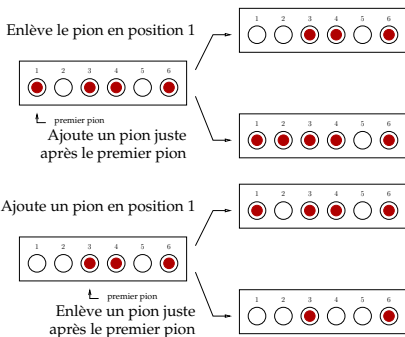
la récursivité s'emmêle...



sciences manuelles du numérique
DÉCONSTRUIRE L'INFORMATIQUE POUR MIEUX LA COMPRENDRE

Les règles

On peut enlever ou placer un pion dans la première position ou dans la position juste après le premier pion de la ligne.

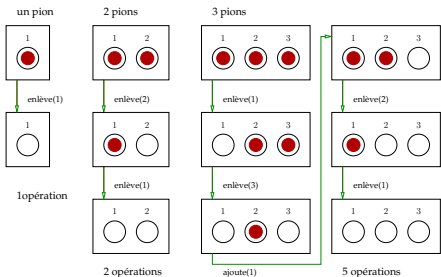


Objectif

Par opérations successives respectant les règles passer d'une ligne pion à une ligne vide.

Premiers pas

Avant de résoudre le problème avec 6 pions, on se *fait la main* avec une ligne plus petite. Pour 1 pion, c'est évident ; pour 2 c'est à peine plus difficile. Pour 3 pions, c'est un peu plus compliqué :



Trouver une méthode générale paraît complexe, car à certaines étapes plusieurs choix sont possibles.

Expression de l'algorithme

L'idée est d'exprimer le problème en fonction de problèmes plus petits, on a remarqué sur les petits exemples que pour résoudre le problème avec 3 pions on utilisait la solution du problème avec 2 pions.

```
Vider ( $n$ )  
    si  $n = 1$  ou  $n = 2$  alors  
1  | C'est facile  
    sinon  
2  | Vider ( $n - 2$ )  
3  | Enlever ( $n$ )  
4  | Remplir ( $n - 2$ )  
5  | Vider ( $n - 1$ )
```

Pour résoudre le problème on suppose que l'on sait résoudre le problème avec une ligne de taille $n - 1$ et $n - 2$ (lignes 5 et 2), on suppose également que l'on sait remplir une ligne de taille $n - 2$ (ligne 4). Évidemment il faut également savoir remplir une ligne vide jusqu'à n . C'est presque le même algorithme.

Récurtivité

```
Remplir( $n$ )  
  si  $n = 1$  ou  $n = 2$  alors  
    └ C'est facile  
  sinon  
    └ Remplir( $n - 1$ )  
      └ Vider( $n - 2$ )  
        └ Ajouter( $n$ )  
          └ Remplir( $n - 2$ )
```

Il reste juste à résoudre le problème pour le **cas de base** (ligne 1) qui terminera les suites d'appels récursifs.

Un tel processus de calcul `Vider` se rappelant lui-même avec un autre paramètre est appelé **récursif**. Ici on a plusieurs appels récursifs dans un même calcul, avec de plus un appel à un autre calcul récursif `Remplir`. C'est un exemple de **récurtivité croisée**.

Exprimer la solution algorithmique sous cette forme améliore la compréhension et rend la programmation plus facile (la plupart des langages de programmation acceptent la définition récursive de fonctions ou de procédures).

Nombre d'opérations

On note v_n et r_n le nombre d'opérations Ajouter ou Enlever pour Vider ou Remplir une ligne de taille n . On a $v_1 = r_1 = 1$, $v_2 = r_2 = 2$, ensuite v_n et r_n sont définis par récurrence :

$$\begin{cases} v_n &= v_{n-2} + 1 + r_{n-2} + v_{n-1}; \\ r_n &= r_{n-1} + v_{n-2} + 1 + r_{n-2}. \end{cases}$$

Après quelques calculs on obtient

$$v_n = r_n = \begin{cases} \frac{1}{3}(2^{n+1} - 2) & \text{si } n \text{ est pair;} \\ \frac{1}{3}(2^{n+1} - 1) & \text{si } n \text{ est impair.} \end{cases}$$

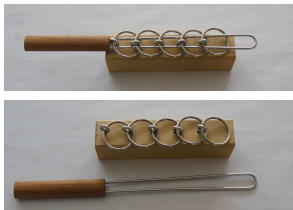
Les premières valeurs sont :

n	1	2	3	4	5	6	7	8	...
v_n	1	2	5	10	21	42	85	170	...

À chaque étape on double environ le nombre d'opérations, on observe ainsi une croissance **exponentielle** et pour n grand le temps nécessaire pour vider la ligne sera extrêmement long.

Le baguenaudier

Ce problème est ancien et se retrouve sous différentes formes, la plus connue est le problème du *baguenaudier* ou *anneaux chinois*, où il faut libérer une "épingle" maintenue par des anneaux. Le mathématicien Jérôme Cardan (1501-1576) est l'un des premiers à étudier ce problème.



En informatique ce type de parcours de codes binaires en ne changeant qu'un seul bit à la fois est très utile pour la conception de circuits logiques (hardware). En 1953, l'ingénieur Franck Gray a ainsi développé un code, portant son nom, utilisé dans de nombreux dispositifs informatiques de codage de signal.

Références

Pour l'introduction à ce problème, et avec d'autres problèmes amusants, on consultera :

Lucas, E. (1891), Récréations mathématiques, Vol. 1, Gauthier-Villars & Fils.

pour la récursivité le chap. 5 du livre :

Dowek, G. et al. (2012), Informatique et Sciences du Numérique, Eyrolles.



Vous pouvez copier, modifier et diffuser librement ce document, à la seule condition de laisser ces mêmes droits à vos lecteurs.
