

# Structures de données et algorithmes

## Répétition 7: Résolution de problèmes

Jean-Michel BEGON

9 décembre 2016

### 100-chaîne

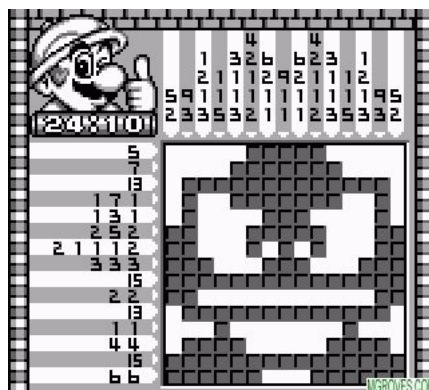
Soit la chaîne "123456789". Proposez un algorithme qui énumère les différentes manières d'insérer des "+" et des "-" entre les *nombres* de manière à obtenir un total de 100.

Par exemple,  $123 + 45 - 67 + 8 - 9 = 100$ .

Quelle est la complexité de votre solution ?

### Logimage - Picross

Proposez un algorithme par recherche exhaustive pour résoudre un logimage :



Cette solution est-elle envisageable en pratique ? Par exemple pour une grille  $15 \times 15$  ?

### Catalan

Dans une répétition précédente, nous avons solutionné l'exercice :

*Soit un ensemble de  $N$  valeurs entières distinctes. Ecrivez une fonction calculant le nombre d'arbres binaires de recherche distincts qu'il est possible de construire à partir de ces  $N$  valeurs ( $N \geq 1$ ).*

par le pseudo-code suivant :

```

NBTREE( $N$ )
1  if  $N \leq 1$ 
2      return 1
3   $Nb = 0$ 
4  for  $i = 1$  to  $N$ 
5       $Nb = Nb + \text{NBTree}(i - 1) * \text{NBTree}(N - i)$ 
6  return  $Nb$ 

```

dont la complexité est importante à cause des appels à des sous-problèmes déjà résolus.

- Dessinez le graphe des sous-problèmes pour une taille de  $N = 4$ .
- Utilisez la mémoïsation pour faire baisser la complexité de l'algorithme (donnez le pseudo-code d'une version ascendante et d'une version descendante).
- Quelles sont ces complexités ?

## 1 Couture minimale (*Checkerboard*)

Mister B. dispose d'une pièce pavée de  $w + 1 \times h$  dalles. A l'exception de la première rangée, chaque dalle est occupée par une pile de dossiers, si bien que la porte de sortie n'est plus accessible. Peu enclin aux travaux domestiques, Mister B. aimerait récupérer l'usage de sa porte tout en minimisant son effort. Il souhaite donc enlever une seule pile par jour et avancer d'une rangée de case à chaque fois (il ne peut enlever que la pile de devant ou celles en diagonales d'une case déjà libérée). Partant du postulat que l'effort est proportionnel à la taille de la pile (et que celle-ci est connue), comment Mister B. peut-il minimiser son effort total ?

					Hauteur/ligne	
	0	1	7	3	2 ✖	1
	0	6	1	3	1	2
	0	4	4	3	5	3
	0	2	9	3	7	4
Largeur/colonne	1	2	3	4	5	

## 2 Plus longue sous-séquence palindromique (adapté de CLRS, 15-2)

On souhaite déterminer le plus grand palindrome qui existe au sein d'un mot. Par exemple, le mot **characters** contient un palindrome de taille 5 : **carac** (il ne s'agit pas d'une sous-séquence contiguë).

- Proposez une solution *brute-force* à ce problème.
- Proposez une solution par programmation dynamique pour ce problème.

## 3 Multiplications matricielles

Lorsqu'on désire multiplier plusieurs matrices  $A_1 \times A_2 \times \dots \times A_n$ , l'ordre des multiplications peut avoir un impact important. Pour trois matrices  $A[m \times n]$ ,  $B[n \times p]$ ,  $C[p \times q]$ , le coût du produit  $(A \times B) \times C$  est de  $\Theta(mnp + mpq)$  alors que le coût du produit  $A \times (B \times C)$  est de  $\Theta(mnq + npq)$ . En fonction des valeurs de  $m, n, p, q$  un des produits est plus avantageux que l'autre.

- (a) Proposez une solution *brute-force* à ce problème.
- (b) Proposez une solution par programmation dynamique pour ce problème.

## 4 Le problème de la partition

On souhaite déterminer s'il est possible de partitionner un tableau d'entiers positifs  $A$  en deux sous-tableaux de somme identique.

- (a) Proposez une solution *brute-force* à ce problème. Quelle est sa complexité ?
- (b) Proposez un algorithme par programmation dynamique pour ce problème. Quelle est sa complexité ?

## 5 Le vrai parenthésage

Soit une expression booléenne composée des symboles `true`, `false`, `and` et `or`. Proposez un algorithme qui détermine le nombre de façons de parenthéser l'expression de sorte qu'elle soit évaluée à `true`.

## 6 Programmation dynamique et données

Quels types de données se prêtent bien à des problèmes de programmation dynamique ? Que représentent les états, quels sont les actions possibles ?

## Bonus

En bioinformatique, l'alignement de séquences est un outil qui permet de représenter deux ou plusieurs séquences d'ADN de manière à en faire ressortir les régions homologues. L'objectif de l'alignement est de disposer les composants de sorte à identifier les zones de concordance.

Par exemple, étant donné les deux séquences de bases :

```
AGGCTATCACCTGACCTCCAGGCCGATGCC
TAGCTATCACGACCGCGGTCGATTTGCCCGAC
```

Leur alignement consiste à appairer une base (un caractère) de l'une soit avec une base de l'autre, soit avec un trou :

```
-AGGCTATCACCTGACCTCCAGGCCGA--TGCCC---
TAG-CTATCAC--GACCGC--GGTCGATTTGCCCGAC
```

Proposer un algorithme par programmation dynamique qui permet d'aligner deux séquences de bases en minimisant le nombre de trous. Etudier sa complexité.