

TD n°3

Programmation dynamique

1 Le triangle de Pascal

On rappelle la récurrence bien connue des coefficients du binôme :

- Pour tout n , $C_n^1 = C_n^n = 1$
- Pour tout n et tout $1 < p < n$, $C_n^p = C_{n-1}^p + C_{n-1}^{p-1}$

Exercice 1 Donnez une fonction récursive calculant C_n^p et calculez combien d'appels récursifs sont faits, dans le pire des cas (plus précisément, pour tout n , pour le pire p)

Exercice 2 Résolvez le problème en utilisant la programmation dynamique.

Indice : la structure de mémorisation que vous utiliserez s'appelle le triangle de Pascal (ou de Tartaglia).

Là aussi, comptez le nombre d'appels récursifs et comparez avec l'exercice précédent.
Quel est la taille de la mémoire utilisée ? Peut-on faire mieux ?

2 Mise en page et impression d'un texte

On dispose d'un fichier texte sans passage à la ligne (il s'agit d'un seul paragraphe) que l'on veut imprimer sur une feuille de taille donnée. Chaque caractère imprimé occupe la même largeur, y compris les espaces, comme sur une machine à écrire mécanique. Le problème est de découper le texte en lignes, pour minimiser les espaces qui restent à la fin de chaque ligne.

Formellement, le texte est une suite de n mots de longueurs l_1, l_2, \dots, l_n , mesurées en nombre de caractères. Chaque ligne contient M caractères, blancs compris. Si une ligne contient les mots i à j (inclus), où $i \leq j$, et qu'on laisse exactement un espace entre deux mots, le nombre de caractères blancs à la fin de la ligne est

$$M - j + i - \sum_{k=i}^j l_k$$

On veut minimiser la somme, sur toutes les lignes hormis la dernière, des cubes des nombres de caractères blancs présents à la fin de la ligne. On dira alors que la présentation du texte est « équilibrée ».

Exercice 3 Donner la solution optimale pour des lignes de longueur 17 avec le texte suivant :

La programmation dynamique est une des bases algorithmiques

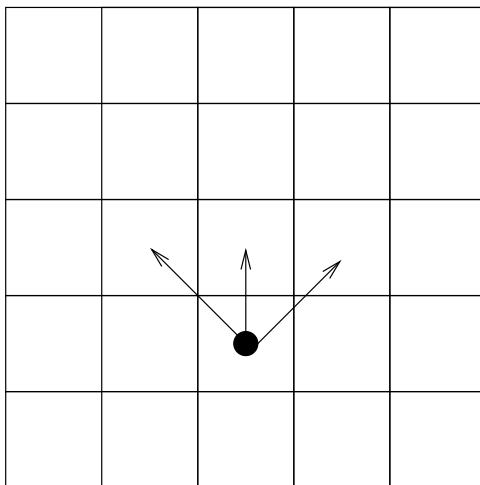
Exercice 4 Montrez que l'algorithme naïf (glouton) échoue.

Exercice 5 Donner un algorithme de programmation dynamique qui résout le problème.

Exercice 6 Donner la complexité en temps et en espace de l'algorithme précédent.

3 Un jeu de dames...

On dispose d'un damier de dimension $n \times n$. A chaque case (i, j) est affecté un coût $c_{i,j}$. On veut déplacer un pion sur le damier depuis n'importe quelle case de la rangée du bas ($i = 1$) vers n'importe quelle case de la rangée du haut ($i = n$). A chaque fois que l'on passe par une case, on paye le coût de cette case. Le but est donc de trouver le chemin de coût minimal, sachant que le pion situé sur la case (i, j) ne peut aller en un coup que sur les cases $(i + 1, j - 1)$, $(i + 1, j)$ ou $(i + 1, j + 1)$. Comme sur le dessin.



Exercice 7 Trouvez une relation de récurrence pour calculer ce qu'il coûte d'atteindre la case (i, j) . Comment gérer les "effets de bords" de manière pratique pour que cette relation reste vraie même pour $j = 1$ ou $j = n$?

Exercice 8 En déduire un algorithme pour résoudre le problème. L'algorithme doit afficher non seulement le coût minimal mais aussi le chemin à suivre.

Exercice 9 Quelle est la complexité de cet algorithme ?