

TP : Programmation dynamique

1 Problème

Une grille de trois lignes et quatre colonnes est représentée ci-dessous :



On souhaite relier le point de départ au point d'arrivée en effectuant des déplacements horizontaux ou verticaux le long des traits dessinant la grille.

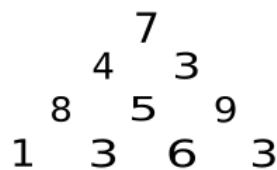
Voici plusieurs pistes d'étude :

- Combien y a-t-il de chemins différents pour relier ces deux points dans le cas de cette grille ?
- Qu'en est-il avec une grille 10 x 10 ?
- Proposer un programme qui prend en compte les dimensions de la grille et renvoie le nombre de chemins reliant ces deux points en effectuant uniquement des déplacements horizontaux et verticaux.
- Proposer une version récursive puis une version en programmation dynamique et comparer leur efficacité.

2 Problème

Sur la page wikipédia présentant la programmation dynamique, l'exemple des nombres pyramidaux est utilisé pour expliquer le principe de la programmation dynamique.

Prenons comme exemple la pyramide de nombres suivante.



L'objectif est de déterminer, en partant du sommet, le chemin jusqu'à la base tel que la somme des nombres traversés soit maximale. Chaque déplacement se fait d'un nombre vers un autre nombre situé juste en dessous, à gauche ou à droite.

Par exemple, un chemin possible pour la pyramide ci-dessus est 7 - 4 - 5 - 3.

Voici plusieurs pistes d'étude :

- Quel est le chemin de cette pyramide qui donne la somme maximale ?
- Proposer un programme qui prend en paramètre une pyramide et renvoie la plus grande somme que l'on peut obtenir et la liste des nombres à traverser. On pourra séparer les résultats dans des fonctions distinctes.
- Proposer une version récursive puis une version en programmation dynamique et comparer leur efficacité.

3 Problème

En bio-informatique, un problème algorithmique essentiel est celui de l'alignement de de séquences pour comparer l'ADN. On peut le formaliser avec des chaines de caractères en mettant en correspondance **le plus possible** les caractères de la première chaine avec les caractères de la seconde chaine en permettant l'insertion de trous dans les chaines de caractères.

Par exemple : On prend les chaines de caractères **GENOME** et **ENORME**. On peut aligner leurs caractères de la façon suivante :

GENO-ME
-ENORME

Les deux tirets représentent des trous.

Il existe plusieurs façons d'aligner ces deux chaines, comme sur l'exemple ci-dessous :

G--ENOME -
-ENOR--ME

La seule contrainte est d'utiliser tous les caractères de chaque chaine, dans l'ordre sans aligner de trous entre eux. Parmi tous les alignements possibles, on cherche celui qui donne le score maximal calculé de la manière suivante :

- l'alignement de 2 caractères identiques : +1
- l'alignement de 2 caractères différents : -1
- le tiret représentant un trou est considéré comme un caractère.

Le premier alignement proposé ci-dessus donne un score de 3 points (5 lettres alignées et 2 lettres non alignées).

Voici plusieurs pistes d'étude :

- On procède à l'alignement de la dernière lettre E des mots **GENOME** et **ENORME**. On a trois cas de figure : E alignés ensemble ou E aligné avec un trou (tiret)
Ensuite on passe au reste des chaines de caractères que l'on essaie d'aligner.
- Proposer une version récursive puis montrer la limite de cette programmation.
- Créer un tableau de scores possibles selon les alignements et s'appuyer dessus pour réaliser une programmation dynamique.