

# TP 8— PROGRAMMATION DYNAMIQUE

<http://www.gallium.inria.fr/~scherer/teaching/colles/2012/tp8.pdf>

Les questions en rapport avec les TP, Caml ou la programmation en général sont les bienvenues et peuvent être envoyées à [gabriel.scherer@gmail.com](mailto:gabriel.scherer@gmail.com). N'hésitez pas !

## 1 TRIANGLE

---

Le problème du triangle opère sur des tableaux-de-tableaux triangulaires de la forme suivante.

```
let triangle =  
  [|  
    [| 1 |];  
    [| 3; 0 |];  
    [| 2; 3; 7 |];  
  |]
```

On considère des “chemins” dans ce triangle, partant de la case du haut (1 dans l'exemple) et allant jusqu'à la dernière ligne. Depuis une case, un chemin peut aller soit à la case immédiatement en-dessous, soit à la case en dessous à droite. Depuis 0 dans l'exemple on peut aller en 3 ou 6, mais pas en 2.

La question de cette section est de calculer la somme maximale des nombres sur un chemin: si on regarde, pour tous les chemins de la première à la dernière ligne, la somme des nombres parcourus, quelle est la plus grande somme possible ?

Dans notre exemple, le meilleur résultat est 8, obtenu par le chemin  $1 \rightarrow 0 \rightarrow 7$ .

▷ **Question 1.** Chercher une description mathématique précise de la fonction qui à un triangle (représenté par une matrice) associe la plus grande somme de chemin. Je m'attends à une relation de récurrence définissant la valeur  $\text{triangle}(M, i, j)$ , et renvoyant la plus grande somme de chemins allant de la case  $(i, j)$  à la dernière ligne de la matrice triangulaire  $M$ . ◀

▷ **Question 2.** Implémenter un programme Caml calculant la plus grande somme d'un triangle fournit en entrée. Quelle est sa complexité en temps et en mémoire ? Peut-on l'améliorer ? ◀

(Ici le chargé de TD explique avec brio les différents choix d'implémentation et comment passer continuellement des uns aux autres.)

## 2 DISTANCE D'ÉDITION

---

*Distance de Levenshtein* : étant donné deux chaînes de caractère  $s_1$  et  $s_2$ , la distance d'édition est le nombre minimal d'opérations permettant de passer d'un mot à l'autre,, parmi les opérations suivantes :

- effacer un des caractères de  $s_1$
- insérer un caractère de  $s_2$  dans  $s_1$

Par exemple, la distance d'édition entre "PREPA" et "BETA" est de 4: on peut supprimer P et R, et ajouter B et T.

▷ **Question 3.** Chercher une description mathématique précise de la fonction qui, étant donné une paire de mots représentés comme des listes de symboles, renvoie leur distance d'édition. Encore une fois, on pourra l'exprimer comme une formule de récurrence. ◀

▷ **Question 4.** En vous servant de la méthode de programmation dynamique expliquée précédemment, écrire un algorithme qui calcule la distance d'édition entre deux mots. Quelle est sa complexité en temps et en mémoire ? ◀

On peut utiliser cet algorithme pour repérer les fautes ou typos et les corriger. Si par exemple vous entrez un nom de variable qui n'existe pas, le compilateur peut regarder parmi les noms existants si certains sont à une petite distance d'édition, et les proposer comme corrections probables. Les versions récentes de OCaml font des suggestions de cette manière.

▷ **Question 5.** Un peu plus difficile : dans triangle, écrire une fonction qui non seulement renvoie la plus grande somme, mais renvoie le chemin correspondant (sous forme d'une liste d'entiers). Pour la distance d'édition, on pourra envisager de renvoyer une des séquences d'édérations optimales. ◀