

Devoir en temps libre n° 5

Ce devoir est à préparer pour le lundi 28 mai 2018 et à rendre en début de TP. Les consignes sont les mêmes que d'habitude. Un modèle de fichier CAML dans lequel quelques fonctions sont déjà implémentées est à télécharger sur le site du cours.

Le but de ce sujet est d'implémenter les exemples d'algorithmes du chapitre sur la programmation dynamique, qu'il faut donc évidemment avoir lus et bien compris.



Nous prendrons la convention^a que les indices commencent à 0 et non à 1 comme dans le polycopié. Il faudra donc faire très attention à ce point et ne pas hésiter à réécrire les algorithmes avant de les implémenter.

^a. Qui est la convention standard en informatique.

On initialisera les tableaux avec la valeur -1 que l'on utilisera également pour représenter ∞ . Vous vérifierez bien que vos fonctions ont exactement le bon type et exactement la même sortie que dans les exemples ci-dessous.

Question 1

Chercher l'exercice 15.1.2.

Question 2

Chercher l'exercice 15.1.3.

Question 3

Compléter le code de la fonction `plus_rapide_chemin` qui prend en entrée a, t, e, x et n , et qui renvoie f^*, l^*, f et l . Pour l'exemple de la figure 15.2, donné juste avant le code à compléter, on doit obtenir :

OCAML

```
let fstar, lstar, f, l = plus_rapide_chemin a t e x n;;
fstar : int = 38
lstar : int = 0
f : int array array = [| [| 9; 18; 20; 24; 32; 35 |];
                        [| 12; 16; 22; 25; 30; 37 |] |]
l : int array array = [| [| -1; 0; 1; 0; 0; 1 |];
                        [| -1; 0; 1; 0; 1; 1 |] |]
```

Question 4

Chercher l'exercice 15.1.1.

Question 5

Implémenter en CAML la fonction `afficher_postes : int -> int array array -> int -> unit` de l'exercice 15.1.1. On aura donc :

OCAML

```
afficher_postes lstar l n;;
chaîne 1, poste 1
chaîne 2, poste 2
chaîne 1, poste 3
chaîne 2, poste 4
chaîne 2, poste 5
chaîne 1, poste 6
- : unit = ()
```

Question 6

Implémenter la fonction `multiplier_matrices : int array array -> int array array -> int array array`.

Question 7

Chercher l'exercice 15.2.3.

Question 8

Implémenter la fonction `ordre_chaine_matrices : int array -> int array array * int array array`. Les cases non utilisées auront pour valeur `-1`. On pourra ensuite vérifier les résultats avec la fonction d'affichage donnée. Par exemple :

OCAML

```
let m, s =
  ordre_chaine_matrices [|30; 35; 15; 5; 10; 20; 25|]
;;
m : int array array =
  [| [|0; 15750; 7875; 9375; 11875; 15125|];
    [| -1; 0; 2625; 4375; 7125; 10500|];
    [| -1; -1; 0; 750; 2500; 5375|];
    [| -1; -1; -1; 0; 1000; 3500|];
    [| -1; -1; -1; -1; 0; 5000|];
    [| -1; -1; -1; -1; -1; 0|]
  |]
s : int array array =
  [| [| -1; 0; 0; 2; 2; 2|];
    [| -1; -1; 1; 2; 2; 2|];
    [| -1; -1; -1; 2; 2; 2|];
    [| -1; -1; -1; -1; 3; 4|];
    [| -1; -1; -1; -1; -1; 4|];
    [| -1; -1; -1; -1; -1; -1|]
  |]

affichage_parenthesage_optimal s;;
((A1(A2A3))((A4A5)A6))
- : unit = ()
```

Question 9

Implémenter la fonction `memorisation_chaine_matrices`.

OCAML

```
memorisation_chaine_matrices [|30; 35; 15; 5; 10; 20; 25|];;
- : int = 15125
```

Question 10

Implémenter la fonction `plus_longue_sous_sequence : string array -> string array -> int array array * string array array`. La matrice `b` sera initialisée avec `"none"` et on utilisera `"up"`, `"diag"` et `"left"` pour les symboles \uparrow , \nearrow et \leftarrow , respectivement.

OCAML

```
let x = [|"a"; "b"; "c"; "b"; "d"; "a"; "b"|];;
let y = [|"b"; "d"; "c"; "a"; "b"; "a"|];;
let c, b = plus_longue_sous_sequence x y;;
c : int array array =
  [| [|0; 0; 0; 0; 0; 0; 0|]; [|0; 0; 0; 0; 1; 1; 1|];
    [|0; 1; 1; 1; 1; 2; 2|]; [|0; 1; 1; 2; 2; 2; 2|];
    [|0; 1; 1; 2; 2; 3; 3|]; [|0; 1; 2; 2; 2; 3; 3|];
    [|0; 1; 2; 2; 3; 3; 4|]; [|0; 1; 2; 2; 3; 4; 4|] |]
b : string array array =
  [| [|"none"; "none"; "none"; "none"; "none"; "none"; "none"|];
    [|"none"; "up"; "up"; "up"; "diag"; "left"; "diag"|];
    [|"none"; "diag"; "left"; "left"; "up"; "diag"; "left"|];
    [|"none"; "up"; "up"; "diag"; "left"; "up"; "up"|];
    [|"none"; "diag"; "up"; "up"; "up"; "diag"; "left"|];
    [|"none"; "up"; "diag"; "up"; "up"; "up"; "up"|];
    [|"none"; "up"; "up"; "up"; "diag"; "up"; "diag"|];
    [|"none"; "diag"; "up"; "up"; "up"; "diag"; "up"|] |]
```

Question 11

Comprendre comment est implémentée la fonction `imprimer_plsc`.

OCAML

```
imprimer_plsc b x y;;
b;c;b;a;
- : unit = ()
```