

Travaux pratiques n°3

Exercices d'algorithmique

F. CUVELLIER, J. TANOI

M.P.S.I. 1, 2018–2019

1 Manipulation de tableaux de nombres

1.1 Recherches de plus grands éléments

Exercice 1. On considère un tableau T d'au moins deux nombres réels.

1. Proposer un algorithme de recherche du plus grand élément de T . Quelle est sa complexité?
2. Compléter cet algorithme pour qu'il donne l'indice le plus petit possible du plus grand élément de T . Quelle est sa complexité?
3. Proposer un algorithme de recherche du deuxième plus grand élément de T (éventuellement égal au plus grand élément de T si ce nombre se trouve plusieurs fois dans T). Quelle est sa complexité?
4. Implémenter ces algorithmes.

1.2 Sous-tableaux croissants maximaux d'un tableau de nombres

Exercice 2. Soit T un tableau non vide de nombres, de longueur n .

1. Proposer un algorithme qui, à partir de T , retourne le tableau I dont les éléments sont les indices non nuls k tels que $T[k-1] > T[k]$, rangés dans l'ordre croissant, et la longueur n de T .
2. Un sous-tableau de T est un tableau formé d'éléments consécutifs de T . Il sera dit *croissant* si ses éléments sont rangés dans l'ordre croissant. Un sous-tableau croissant sera dit *maximal* s'il n'est contenu dans aucun sous-tableau croissant autre que lui-même.
Écrire un algorithme qui, à partir de T , donne le tableau formé des sous-tableaux croissants maximaux de T .
3. Proposer un algorithme qui retourne la plus grande des longueurs des sous-tableaux croissants maximaux de T .
4. Implémenter ces algorithmes.
5. Le module `random` contient une fonction `randint()` qui produit des nombres entiers aléatoires : `random.randint(a, b)` donne un nombre entier supérieur ou égal à a et inférieur ou égal à b .
Former mille tableaux de cents nombres entiers compris entre 1 et 1000, pour chacun d'eux, déterminer la plus grande des longueurs des sous-tableaux croissants maximaux, et donner le tableau ordonné des nombres obtenus.

2 Manipulations de tableaux rectangulaires

2.1 Tableaux rectangulaires

Un tableau T est dit *rectangulaire* si ses éléments sont des tableaux, tous de même longueur. Par abus, les éléments des tableaux qui sont les éléments de T seront encore appelés *éléments* de T .

Dans les questions suivantes, les valeurs des éléments d'un tableau rectangulaire importent peu. Elles pourraient être des nombres, des chaînes de caractères, etc. Pour simplifier, on décidera que ce sont des chaînes formées d'un seul caractère (qui pourrait représenter une couleur).

Exercice 3. On veut visualiser un tableau rectangulaire sous la forme d'une chaîne de caractères rectangulaire (en apparence). Écrire une fonction en Python, qui transforme un tableau rectangulaire T en la chaîne de caractères obtenue par concaténation des éléments du tableau, de sorte que deux lignes consécutives soient séparées par un caractère de nouvelle ligne (`\n`).

2.2 Éléments contigus, éléments connectés

Exercice 4. On dit que deux éléments d'un tableau rectangulaire T sont *contigus* s'ils se trouvent l'un à côté de l'autre, sur une ligne ou sur une colonne; autrement dit, ils sont contigus si, pour des entiers convenables h et k , leurs couples d'indices sont ou (h, k) et $(h + 1, k)$, ou (h, k) et $(h, k + 1)$.

Deux éléments de T sont dits *connectés* s'ils sont l'un le premier, l'autre le dernier élément d'une suite d'éléments de T , dont deux consécutifs quelconques sont identiques et contigus.

1. Donner un algorithme qui, à partir d'un tableau rectangulaire T et d'un couple d'indice (a, b) , retourne l'ensemble des éléments de T connectés à l'élément d'indice (a, b) (qui, en Python, sera `T[a][b]`).
2. Implémenter cet algorithme.
3. Écrire une fonction qui remplace le caractère d'un élément t de T et tous les éléments connectés à t par un caractère c donné.

2.3 Jeu de la vie (d'après John Conway)

Les éléments d'un tableau rectangulaire T , à m lignes et n colonnes, sont des chaînes d'un seul caractère, qui est ' ' (case vide) ou 'x' (case pleine).

Un élément de T est repéré par ses indices h et k . Par définition, ses *voisins* seront les éléments autres que lui, dont le premier indice est $h - 1$, h ou $h + 1$ et le second indice $k - 1$, k ou $k + 1$ (quand c'est possible).

Exercice 5. À un élément de T , on associe le *voisinage*, qui est le nombre de ses voisins qui sont des cases pleines (c'est-à-dire dont la valeur est "x").

Écrire une fonction qui, à partir d'un tableau T et à des indices h et k , retourne le voisinage de l'élément de T d'indice (h, k) .

Exercice 6. Un tableau rectangulaire T se transforme un tableau rectangulaire U de la façon suivante :

- Une case vide de T devient une case pleine de U si son voisinage vaut 3. Sinon, elle reste vide.
 - Une case pleine de T devient une case vide de U si son voisinage vaut 0, 1, ou 4 ou plus. Sinon, elle reste pleine.
1. Écrire une fonction qui à partir d'un tableau T retourne le tableau U .
 2. Écrire une fonction qui visualise l'évolution d'un tableau T au cours de n transformations successives.