MPCI

Programmation 2

Examen Final

Mercredi 4 mars 2020

On rappelle qu'aucun document ni équipement électronique n'est autorisé. L'usage du thermomètre à mercure est néanmoins toléré. La clarté & la concision des réponses sera appréciée. Les algorithmes doivent **tous** *être prouvés* **et** *donnés avec leur complexité.*

L'examen est composé de 5 exercices indépendants valant chacun 4 pts.

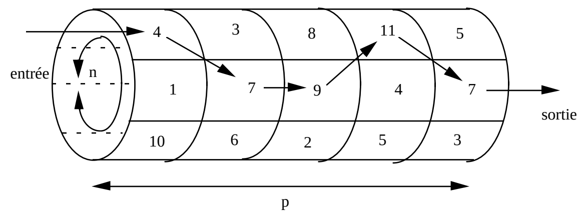
Exercice 1 : On souhaite recherche si un entier `e` est dans une liste triée d'entiers `L`. Ecrivez un algorithme utilisant la méthode dichotomique pour résoudre ce problème.

Exercice 2 : Un *col* dans une matrice est un élément qui est le plus petit de sa colonne & le plus grand de sa ligne.

1. Est-ce que toute matrice admet un col ?
2. Est-ce que certaines matrices admettent un col ?
3. Ecrivez un algorithme qui détermine si une matrice admet un col & qui, dans le cas positif, donne ses coordonnées (ligne & colonne).

Exercice 3 : Un morceau (musique, vidéo, ...) est représenté par un objet d'une classe `Morceau` précisant son titre et sa durée. Ces deux informations seront transmises au moment de la construction de l'objet.

1. Donnez le code python de la classe `Morceau`.
2. Ecrivez en python un programme principal (c'est à dire que l'on suppose la classe Morceau déjà écrite) qui crée un Morceau dont le titre est *Stupeflip, vite !* d'une durée de 3 minutes 39.
3. Une `Playlist` est composée d'une liste de Morceaux de musique ou de vidéos. On créera une Playlist en précisant une liste de Morceaux. Donnez le code python de cette classe.
4. Ajoutez une méthode nommée `morceau` à la classe `Playlist` qui rend le morceau dont le nom est passé en paramètre (ou `None` si le morceau n'existe pas).



Exercice 4 : On considère $n.p$ entiers positifs a_{ij} ($0 \leq i < n, 0 \leq j < p$), écrits sur un cylindre ayant n lignes & p colonnes, comme illustré ci-dessous. Un chemin est tracé de l'entrée du cylindre jusqu'à la sortie, avec la restriction que, d'une case, on ne peut aller qu'aux trois positions de la colonne suivante adjacentes à la position courante. Le coût d'un tel chemin est la somme des entiers écrits dans les cases traversées (par exemple, le chemin tracé sur le dessin a un coût égal à 38).

1. Combien de chemins distincts a-t-on de l'entrée à la sortie, les cases de départ & d'arrivée n'étant pas imposées ?
2. Donner un algorithme récursif qui détermine un tel chemin, de coût minimum On justifiera le fait que l'algorithme calcule bien ce qu'il faut, & on donnera sa complexité.
   Indication : Un chemin arrivant à la case $a_{i,j}$ passe forcément par une des cases $a_{i,j-1}$, $a_{i-1,j-1}$ ou $a_{i+1,j-1}$ (les opérations sur i sont faites modulo n).
3. Transformer cet algorithme en un algorithme itératif de complexité $O(np)$.
4. L'appliquer à l'exemple de la figure où l'on suppose que l'on voit tout le cylindre ($n = 3$, $p = 5$) ; en déduire un chemin de coût minimum.

Exercice 5 : La suite de Fibonacci est définie par $F_0 = 0$, $F_1 = 1$, & $F_{i+2} = F_{i+1} + F_i$ pour $i \in N$

1. Donner un algorithme récursif & un algorithme itératif pour calculer F_i.
2. Comparer ces deux algorithmes.