

Algorithmique

Travaux dirigés : feuille 2

Objectifs pédagogiques et méthode de travail. Cette feuille comporte deux parties.

- La partie 1 “Objectif 10” propose des exercices très proches du cours et des révisions du semestre 1. Avec les QCM en ligne (ENT), cette partie a pour objectif l’assimilation des connaissances de base, en particulier tout ce qui a été vu au semestre 1. Ces notions pleinement acquises permettent d’obtenir la moyenne lors des contrôles de connaissances (contrôle continu et sessions d’examens).
- La partie 2 “Objectif 20” propose des exercices de difficulté croissante sur des notions plus avancées ou des formulations plus abstraites. Les étudiant-e-s informaticien-e-s, et les étudiant-e-s mathématicien-e-s qui souhaitent continuer au delà de la Licence, doivent impérativement traiter cette partie 2.

1 Connaissances

- tableaux 2D et traitements associés : images, matrices
- boucles imbriquées
- encore un petit peu de complexité en temps

2 Objectif 10

Exercice 1.

1. Dérouler l’algorithme suivant en explicitant les valeurs de l’ensemble de ses variables.

```
1 declare
.   i, j, k : entier = 0
.   debut
.   pour i de 0 à 3 faire
5     pour j de i à 3 faire
.       k = i+j
.     finpour
.   finpour
10 fin
```

2. Même question en remplaçant la ligne 5 par :
pour j de i+1 à 3 faire
3. Compter le nombre d’additions dans chacun des deux cas.

Exercice 2.

1. Écrire un algorithme qui calcule les tables de multiplications entre entiers compris en 1 et 10, les stocke dans un tableau 2D puis les affiche. Ainsi la case (i, j) du tableau contient le résultat de $i \times j$.

Exercice 3.

On suppose l’existence d’une procédure qui initialise un tableau 2D d’entiers à partir d’une variable `img` qui contient une image bitmap (NB ou à niveaux de gris). Son en-tête est :

```
procedure LireImage(img: in image , l: in entier, c: in entier, t: out tableau[l,c] d’entiers)
```

Les images sont déclarées et initialisées comme suit et ne sont exploitables que par la procédure `LireImage` : tout traitement doit être réalisé sur le tableau 2D d’entiers obtenu par `LireImage`.

```

declare
  visageNB : image[512, 512] = un_fichier_bitmap_NB
  visageG  : image[512, 512] = un_fichier_bitmap_Gris
  tab : tableau[512, 512] d'entier
debut
  LireImage(visageNB, 512, 512, tab) // tab contient les valeurs des pixels de visageNB
  ...
fin

```

Ecrire les algorithmes des traitements suivants. On pourra commencer en introduisant un tableau supplémentaire pour l'image transformée. Selon les cas, on essaiera ensuite une solution "en place" : la transformation s'effectue sur le tableau de l'image d'origine.

1. Générer le négatif (reverse vidéo) d'une image NB ou par niveau de gris.
2. Générer une image NB à partir d'une image niveau de gris.
3. Augmenter le contraste de la transformaton précédente.
Principe : fixer un seuil et remplacer les pixels plus clairs que le seuil par des pixels blancs, et inversement les plus sombres que le seuil par des pixels noirs.
4. Générer une image miroir vertical ou horizontal d'une image NB.
5. Augmenter la luminosité (ou luminance) d'une image à niveau de gris.
Principe : ajouter ou retrancher une constante de la valeur des pixels.
6. Générer les contours significatifs d'une image.
Principe : on remplace par un pixel noir chaque pixel dont la variation des valeurs de ses 4 voisins varient au delà d'un certain seuil, sinon on le remplace par un pixel blanc.
7. Réduire par 2 la taille d'une image à niveau de gris.
Principe : chaque carré de 2x2 pixels est remplacé par 1 pixel de valeur la moyenne des pixels du carré.

Exercice 4.

Ecrire les algorithmes de vérification suivants pour une matrice M donnée, carrée de taille n et à valeurs flottantes.

1. M est diagonale ?
2. M est symétrique ?
3. M est égale à l'identité ?
4. M est l'inverse d'une autre matrice donnée N ?

Ecrire ensuite le même traitement sous forme de l'appel à une fonction (en-tête, puis appel, puis corps).

Exercice 5.

Ecrire les algorithmes des traitements suivants pour une matrice M donnée, carrée de taille n et à valeurs flottantes.

1. Générer, puis afficher, la matrice transposée de M
2. Générer la matrice symétrique à partir de la partie triangulaire supérieure de M
3. Calculer les opérations matricielles suivantes en veillant à la cohérence des dimensions :
 - (a) addition et produit de matrices : $A + B$, $A \times B$
 - (b) produit matrice-vecteur : $A \cdot U$
4. Compter le nombre d'opérations arithmétiques de ces traitements et préciser la complexité des algorithmes.

Exercice 6.

Calculer, puis afficher, les 15 premières lignes du triangle de Pascal.

3 Objectif 20

Exercice 7.

On reprend l'exercice 1 en introduisant un nombre arbitraire n d'itérations de chaque boucle.

1. Quelle est une mesure naturelle de la complexité de cet algorithme ? Quel est le paramètre de cette complexité ?
2. Pour chacun des cas suivants : compter le nombre d'additions effectuées à chaque itération de la boucle intérieure. En déduire le nombre totale d'additions. Quelle est la complexité asymptotique de cet algorithme ?

(a) On modifie la ligne 4 :

```
4   pour i de 1 à n faire
5     pour j de 0 à 3 faire
```

(b) on modifie maintenant la ligne 5 :

```
4   pour i de 0 à 3 faire
5     pour j de 1 à n faire
```

(c) ou aussi :

```
4   pour i de 0 à 3 faire
5     pour j de i à n faire
```

(d) avec les lignes 4 et 5 suivantes :

```
4   pour i de 1 à n faire
5     pour j de 1 à n faire
```

(e) et aussi les lignes 4 et 5 suivantes :

```
4   pour i de 1 à n faire
5     pour j de i à n faire
```

Rappel. Il est utile de connaître la valeur de la somme des n premiers entiers et son équivalent asymptotique :

$$1 + 2 + \dots + n = \sum_{k=1}^n k = n(n+1)/2 \approx n^2/2 = \theta(n^2).$$

Exercice 8.

Ecrire les algorithmes qui, à partir de l'image originale (à gauche), génèrent les images zébrées horizontales ou verticales ou quadrillées (à droite). Le nombre de découpages horizontaux ou verticaux sont des paramètres de l'algorithme. On veillera en particulier au cas où la taille de l'image ne soit pas divisible par ces paramètres.



Exercice 9.

Ecrire l'algorithme qui résout le système linéaire $Tx = b$ où la matrice T est une matrice triangulaire inférieure (carrée de taille n) et b un vecteur second membre (de taille n). La matrice T et le vecteur b sont supposés donnés (au clavier ou avec une valeur d'initialisation).