

## Algorithmique 1

### Feuille 1 : Rappels

#### Exercice 1.1

On considère la fonction suivante :

```
fonction mystere (ref T:tableau[1..N] d'entier):vide;
var i, j : entier;
debut
  i= 1;
  j= N;
  tant que i < j faire
    si T[i] == 0 alors
      i= i+1;
    sinon
      echanger(T[i], T[j]);
      j= j-1;
    fsi;
  ftq;
fin;
```

1. Que fait-elle en supposant que le tableau T ne contient que des 0 et des 1 ?
2. Modifier l'algorithme de façon à l'appliquer à des tableaux contenant au plus trois valeurs différentes : par exemple 0, 1 et 2 (rappel : chaque élément de T doit être examiné au maximum une fois).  
Peut-on l'étendre aisément à K valeurs différentes ?
3. Donner une version récursive de l'algorithme général.
4. Ecrire une procédure séparant un tableau d'entiers T[1..N] contenant une valeur X en 2 zones : T[1..I-1] contenant les entrées strictement inférieures à X, T[I..N] les entrées supérieures ou égales à X, de façon que X se trouve en position I à la fin du traitement. Etudier l'occupation en mémoire.  
Donner le principe d'un algorithme de tri utilisant cette dernière procédure.

#### Exercice 1.2 *Suite de Fibonacci*

On rappelle que la suite de Fibonacci, 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ... s'obtient de la façon suivante : à partir du rang 3, chaque terme est la somme des deux précédents.

Pour calculer le  $n$ -ième terme de la suite donner une fonction :

- récursive
- récursive terminale
- itérative

Calculer la complexité du calcul des nombres de Fibonacci par les algorithmes récursif et itératif précédents. Il est utile de savoir que les nombres de Fibonacci sont étroitement liés au nombre d'or  $\phi$  et à son conjugué  $\bar{\phi}$ , qui sont donnés par les formules suivantes :

$$\phi = \frac{1 + \sqrt{5}}{2} = 1,61803\dots$$

et

$$\bar{\phi} = \frac{1 - \sqrt{5}}{2} = -0,61803\dots$$

Plus précisément, si on appelle  $F(n)$  le  $n$ -ième nombre de Fibonacci ( $n \geq 0$ ), on a

$$F(n) = \frac{\phi^n - \bar{\phi}^n}{\sqrt{5}}$$

(ce qu'on peut prouver par récurrence).

### Exercice 1.3

Ecrire un algorithme calculant le  $n$ -ième nombre de Fibonacci en temps  $\log_2(n)$ .

Indications :

1. Construire le vecteur  $F\vec{I}B\vec{O}(n) = \begin{pmatrix} F(n) \\ F(n-1) \end{pmatrix}$  et exprimer  $F\vec{I}B\vec{O}(n)$  en fonction de  $F\vec{I}B\vec{O}(n-1)$  à l'aide d'une matrice de dimension  $2 \times 2$  à déterminer.

2. Calculer les puissances d'une matrice en utilisant la méthode alexandrine<sup>1</sup>

### Exercice 1.4

On voudrait stocker un ensemble de cercles de dimension quelconque, dans un premier temps dans l'ordre de leur stockage, ensuite de manière ordonnée (du plus petit au plus grand). Pour rappel, un cercle est la donnée d'un point et d'un rayon. On pourra utiliser le type abstrait `containeur` et les primitives correspondantes comme défini en cours pour représenter et manipuler l'ensemble de cercles.

- Proposer une implémentation permettant une manipulation aisée d'un ensemble de cercles (structures de données adaptées et primitives) en s'inspirant de l'exemple des nombres complexes vu en cours.
- Ecrire les fonctions suivantes :
  1. Qui teste l'égalité de deux cercles
  2. Qui teste l'inclusion d'un cercle dans un autre

### Problème récurrent (notre fil d'ariane)

#### Exercice 1.5 Gestion d'une piste d'atterrissage des avions

Un avion est un enregistrement contenant :

- l'indicatif (6 caractères)
- la destination (30 caractères)
- l'autonomie résiduelle de carburant comptée en heures de vol (entier)
- deux booléens indiquant s'il y a un pirate à bord et s'il y a le feu.

1. Définir les structures de données nécessaires.
2. Ecrire la fonction `Priorité` ainsi que la gestion complète de la piste.
3. Envisager le cas de suppression d'un élément quelconque de la file lorsque le pirate a mis sa menace de détournement à exécution.

Quelles sont les notions que vous venez de voir qui peuvent permettre d'amorcer le fil d'ariane ?

---

1. Pour multiplier deux entiers  $X$  et  $Y$ , on détermine la suite  $(x_0, y_0) = (X, Y)$  et  $(x_{n+1}, y_{n+1}) = (2x_n, \frac{y_n}{2})$ . Le produit  $XY$  est égal à la somme des  $x_i$  pour lesquels  $y_i$  est impair.

**Exercice à faire à la maison****Exercice 1.6**

Ecrire un algorithme calculant le  $n$ -ième nombre de Fibonacci en temps  $\log_2(n)$  utilisant les relations suivantes :

$$F(2k) = F(k)^2 + 2F(k)F(k-1)$$

$$F(2k+1) = F(k)^2 + F(k+1)^2$$

**Exercice 1.7**

On gagne encore en efficacité en retravaillant les relations de récurrence pour éviter les calculs redondants :

$$F(2k) = F(2k+1) - F(2k-1) = (F(k)^2 + F(k+1)^2) - (F(k-1)^2 + F(k)^2) = F(k+1)^2 - F(k-1)^2$$

$$F(k) = 0, k = 0$$

$$F(k) = 1, k = 1, 2$$

$$F(k) = F(\lfloor \frac{k}{2} \rfloor + 1)^2 - (-1)^k F(\lfloor \frac{k-1}{2} \rfloor)^2, k > 2$$

Ecrire un algorithme calculant le  $n$ -ième nombre de Fibonacci en utilisant ces expressions.