

Feuille 1 : Rappels et compléments

Exercice 1.1

On considère la fonction suivante :

```
fonction separer (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 (chaque élément de T doit être examiné au maximum une fois). Peut-on étendre aisément l'algorithme à K valeurs différentes ?
3. Écrire une fonction 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. Donner le principe d'un algorithme de tri utilisant cette dernière fonction.

Exercice 1.2 *Suite de Fibonacci.*

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

1. une fonction récursive ;
2. une fonction récursive terminale ;
3. une fonction itérative.

Calculer la complexité du calcul des nombres de Fibonacci par chacun de ces algorithmes.

Il est utile de savoir que les nombres de Fibonacci sont étroitement liés au *nombre d'or*

$$\varphi = \frac{1 + \sqrt{5}}{2} = 1,61803\dots \text{ et à son conjugué } \bar{\varphi} = \frac{1 - \sqrt{5}}{2} = -0,61803\dots$$

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

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

(ce qu'on peut prouver par récurrence).

Exercice 1.3

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

Indications :

1. Exprimer le vecteur $\overrightarrow{FIBO}(n) = \begin{pmatrix} F(n) \\ F(n-1) \end{pmatrix}$ en fonction de $\overrightarrow{FIBO}(n-1)$ à l'aide d'une matrice de dimension 2×2 à déterminer.
2. Utiliser le calcul logarithmique des puissances d'une matrice.

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). Un cercle est caractérisé par un point et un rayon.

1. 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. On pourra utiliser le type abstrait `containeur` et les primitives correspondantes comme défini en cours
2. Écrire une fonction qui teste l'égalité de deux cercles.
3. Écrire une fonction qui teste l'inclusion d'un cercle dans un autre.

Problème récurrent

Exercice 1.5 Gestion d'une piste d'atterrissage

Un avion est caractérisé par un enregistrement contenant :

- un indicatif (6 caractères)
- sa destination (30 caractères)
- son 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.

Le problème consiste à

1. définir les structures de données nécessaires à la gestion d'une piste d'atterrissage ;
2. définir et écrire une fonction calculant la priorité d'un avion pour l'utilisation de la piste ;
3. définir et écrire les fonctions nécessaires à la gestion complète de la piste (on envisagera la suppression d'un avion piraté de la file d'attente lorsque le pirate a mis sa menace de détournement à exécution).

Quelles notions vues dans ce TD peuvent permettre d'amorcer la résolution du problème ?