1ALGO Algorithmique et logique

Gaël Roustan (Argonaultes)

2025

Abstract

TD 5

TD 5

Suite de Fibonacci

La suite de Fibonacci est définie par

```
f(n) = 0 si n = 0
f(n) = 1 si n = 1
f(n) = f(n-1) + f(n-2) si n >= 2
```

Écrire une fonction récursive nommée Fibonacci permettant de calculer le énième terme de la suite de Fibonacci.

ALGORITHME: suit de fibonacci

```
FONCTION: Fibonacci(ENTREE : n : ENTIER) : ENTIER
DEBUT
    SI (n = 0) ALORS
        RETOURNER: 0
    SINON
        SI (n = 1) ALORS
            RETOURNER: 1
        SINON
            RETOURNER: Fibonacci(n-1) + Fibonacci(n-2)
        FINSI
    FINSI
FIN
```

```
ALGORITHME: suit de fibonacci

FONCTION: Fibonacci(ENTREE : n : ENTIER) : ENTIER
DEBUT
    SI (n = 0 OU n = 1) ALORS
        RETOURNER: n
    SINON
        RETOURNER: Fibonacci(n-1) + Fibonacci(n-2)
    FINSI
FIN
```

Somme des premiers cubes

Écrire une fonction récursive nommée SommeCubes qui calcule la somme des n premiers cubes d'entiers.

```
ALGORITHME: somme des premiers cubes
FONCTION: SommeCubes(ENTREE : n : ENTIER) : ENTIER
DEBUT
    SI (n = 0) ALORS
        RETOURNER: 0
    SINON
        RETOURNER: SommeCubes(n - 1) + n^3
    FINSI
FIN
```

Calcul du PGCD de deux entiers

Écrire une fonction récursive nommée nommé PGCD calculant le PGCD de deux entiers naturels.

PGCD : Plus Grand Commun Diviseur

```
FONCTION: PGCD(ENTREE : a, b : ENTIER) : ENTIER
DEBUT
    SI (b = 0) ALORS
        RETOURNER: a
    SINON
        RETOURNER PGCD(b, MOD(a, b))
FIN
```

Recherche d’un élément dans un tableau

Écrire une fonction récursive nommée Recherche indiquant si un réel donné est présent ou non dans un tableau de réels donné.

```
ALGORITHME: recherche
FONCTION: Recherche(TABLEAU : tab[] : REELS, element : REEL, position_recherche : ENTIER, longueur_tableau : ENTIER) : BOOLEEN
DEBUT
    SI (tab[position_recherche] = element) ALORS
        RETOURNER: Vrai
    SINON
        SI (position_recherche >= longueur_tableau) ALORS
            RETOURNER: Faux
        SINON
            RETOURNER: Recherche(tab, element, position_recherche + 1, longueur_tableau)
        FINSI
    FINSI
FIN
PROGRAMME PRINCIPAL:
VARIABLES:
    element_recherche : REEL
    TABLEAU: montableau[0..4] <-- { 2.1, 3.2, 4.1, 4.2, 6.7 } : REEL
DEBUT
    element_recherche <-- LIRE()
    SI (Recherche(montableau, element_recherche, 0, 5)) ALORS
        ECRIRE("Element trouvé")
    SINON
        ECRIRE("Element introuvable")
    FINSI
FIN
```


Argonaultes, RCS PARIS 978 665 909, NAF : 85.59A
SARL au capital de 3 000 euros,
enregistrée sous le numéro 11756794275 auprès du préfet de région d’Ile-de-France

Recherche dichotomique d’un élément dans un tableau trié

La dichotomie est un principe de traitement consistant à diviser un problème en deux, afin de se ramener à des situations de complexité moindre.

Argonaultes, RCS PARIS 978 665 909, NAF : 85.59A
SARL au capital de 3 000 euros,
enregistrée sous le numéro 11756794275 auprès du préfet de région d’Ile-de-France


Version itérative

Écrire une fonction itérative nommée `rechercheDichotomiqueIterative` utilisant une méthode dichotomique pour tester si un réel saisi par un utilisateur est présent ou non dans un tableau de réels supposé trié.

Remarque : Le tableau est divisé en deux sous parties, puis encore en deux et ainsi de suite jusqu'à pouvoir conclure aisément.

```
ALGORITHME: rechercheDichotomiqueIterative
FONCTION: rechercheDichotomiqueIterative(element : REEL, TABLEAU: tab[] : REELS, longueur: ENTIER) : BOOLEEN
VARIABLES:
    debut, fin, milieu : ENTIERS
DEBUT
    debut <-- 0
    fin <-- longueur - 1
    milieu <-- DIV(debut + fin, 2)

    TANTQUE (debut < fin ET tab[milieu] <> element) ALORS
        SI (tab[milieu] < element) ALORS
            debut <-- milieu + 1
        SINON
            fin <-- milieu - 1
        FINSI
        milieu <-- DIV(debut + fin, 2)
    FINTANTQUE
    RETOURNER tab[milieu] = element
FIN
```

Argonaultes, RCS PARIS 978 665 909, NAF : 85.59A
SARL au capital de 3 000 euros,
enregistrée sous le numéro 11756794275 auprès du préfet de région d'Ile-de-France

Version récursive

Écrire une fonction récursive nommée `rechercheDichotomiqueRecursive` utilisant une méthode dichotomique pour tester si un réel saisi par un utilisateur est présent ou non dans un tableau de réels supposé trié.

```
ALGORITHME: rechercheDichotomiqueRecursive
FONCTION: rechercheDichotomiqueRecursive(element : REEL, TABLEAU: tab[] : REELS, debut, fin: ENTIERS) : BOOLEEN
VARIABLES:
    milieu : ENTIER
DEBUT
    milieu <-- DIV(debut + fin, 2)

    SI (debut > fin) ALORS
        RETOURNER : Faux
    SINON
        SI (tab[milieu] = element) ALORS
            RETOURNER : Vrai
        SI (tab[milieu] < element) ALORS
            RETOURNER : rechercheDichotomiqueRecursive(element, tab, milieu + 1, fin)
        SINON
            RETOURNER : rechercheDichotomiqueRecursive(element, tab, debut, milieu - 1)
        FINSI

    FINSI
FIN
```

Argonaultes, RCS PARIS 978 665 909, NAF : 85.59A
SARL au capital de 3 000 euros,
enregistrée sous le numéro 11756794275 auprès du préfet de région d'Ile-de-France

Algorithmes de tri

Principe : trier par ordre croissant les éléments d'un tableau de nombres.

Une procédure d'échange

Écrire une procédure nommer échanger réalisant la permutation des valeurs de deux variables dans un tableau distribué.

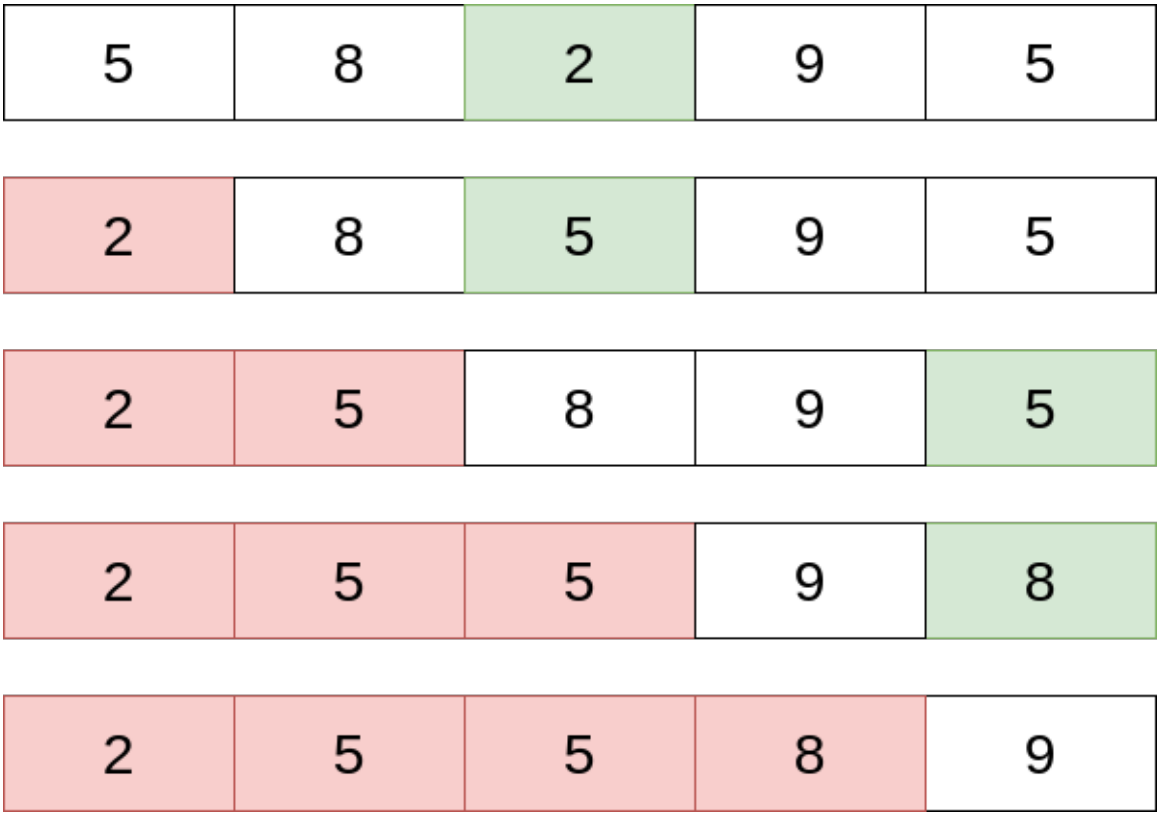
```
ALGORITHME: tri par selection

PROCEDURE: echanger(ENTREE : indice1, indice2 : ENTIERS, ENTREE/SORTIE : TABLEAU : tab[] : REELS)
VARIABLE:
    temporaire : REEL
DEBUT
    temporaire <-- tab[indice1]
    tab[indice1] <-- tab[indice2]
    tab[indice2] <-- temporaire
FIN
```

Tri par sélection

Nous cherchons dans le tableau la plus petite valeur et nous la permutons avec le premier élément du tableau. Nous cherchons ensuite la plus petite valeur à partir de la deuxième case et nous la permutons avec le second élément du tableau. Et ainsi de suite jusqu'à avoir parcouru tout le tableau.

En rouge figurent les éléments déjà triés et en vert le minimum de ceux à trier.



Écrire l'algorithme réalisant le tri par sélection d'un tableau de n entiers.

Ecrire 2 fonctions. Une fonction indiceDuMinimum et une fonction triSelection

Argonaultes, RCS PARIS 978 665 909, NAF : 85.59A
SARL au capital de 3 000 euros,
enregistrée sous le numéro 11756794275 auprès du préfet de région d'Ile-de-France


```
FONCTION: indiceDuMinimum(ENTREE : TABLEAU : tab[] : REEL, indice_debut, longueur : ENTIER) : ENTIER
VARIABLES:
    indice, indice_min : ENTIERS
DEBUT
    indice_min <-- indice_debut
    POUR indice ALLANT DE indice_debut + 1 A longueur - 1 AU PAS DE 1 FAIRE
        SI ( tab[indice] < tab[indice_min] ) ALORS
            indice_min <-- indice
        FINSI
    RETOURNER :  indice_min
FIN

PROCEDURE : triSelection(ENTREE/SORTIE : TABLEAU : tab[] : REEL, ENTREE : longueur_tableau : ENTIER)
VARIABLES :
    indice_courant, indice_min : ENTIERS
DEBUT
    POUR indice_courant ALLANT DE 0 A longueur_tableau - 1 AU PAS DE 1 FAIRE
        indice_min <-- indiceDuMinimum(tab, indice_courant)
        echanger(indice_courant, indice_min, tab)
    FINPOUR
FIN
```