

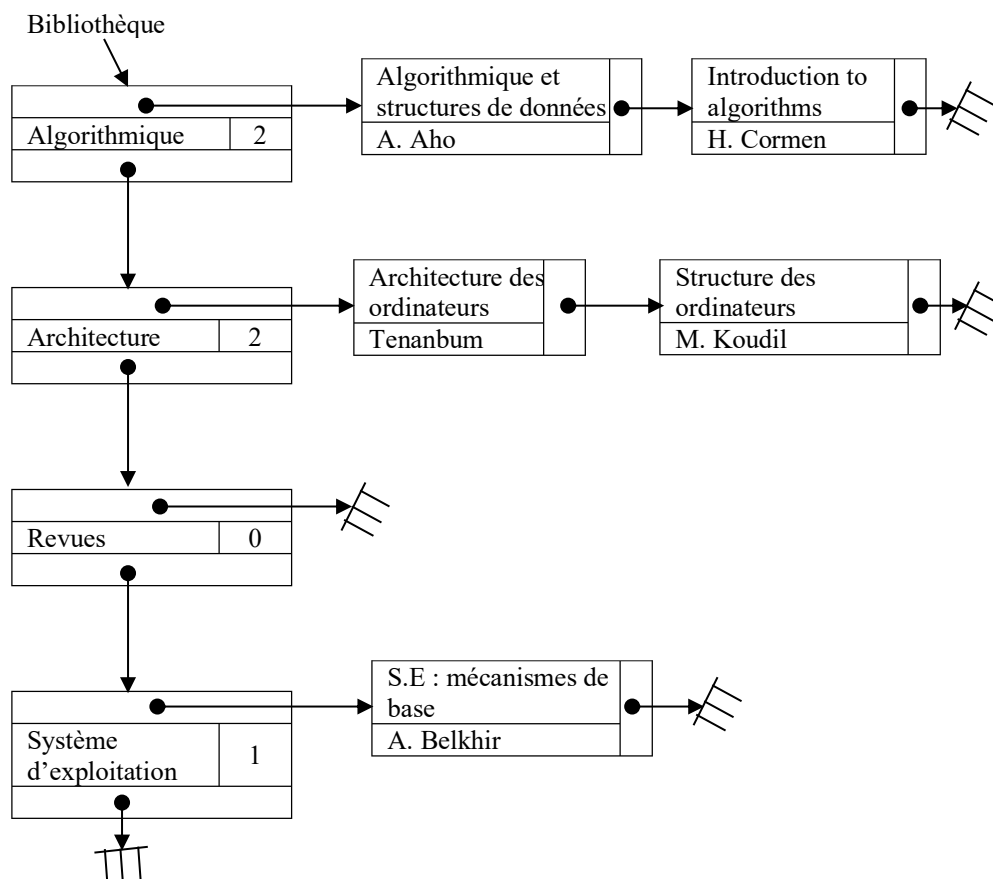
Examen d'algorithmique 1

14h-15h30

A1

Exercice 1 (10 pts)

Dans cet exercice, un étudiant en informatique souhaite représenter sa bibliothèque personnelle en utilisant une structure dynamique. La structure proposée est représentée dans la figure suivante :



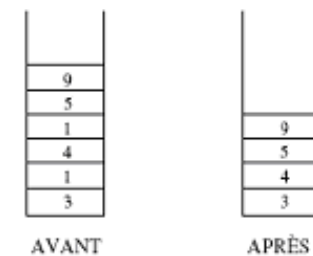
La liste verticale contient les catégories des livres avec le nombre de livres dans chacune, tandis que les listes horizontales contiennent les titres des livres avec leurs auteurs dans chaque catégorie.

1. Donner la déclaration des structures de données nécessaires à l'implémentation de cette bibliothèque ainsi que la procédure d'initialisation de ces structures. 1.5
2. Ecrire la procédure d'ajout d'une nouvelle catégorie (l'ajout se fait toujours à la fin de la liste). 1.5
3. Ecrire la procédure d'insertion d'un nouveau livre (l'ajout se fait au début de la liste) 2
4. Ecrire la procédure qui permette d'afficher les livres d'une catégorie donnée. 1.5
5. Ecrire la fonction qui retourne le nombre total de livre dans la bibliothèque. 1
6. Ecrire la procédure qui permette de supprimer une catégorie avec tous ses livres 2.5

Exercice 2 (5 pts)

1. Les listes (1, 6, 1, 6, 7, 9, 1, 10, 6) et (1, 3, 1, 4, 3, 2, 1, 4, 2) peuvent-elles représenter un tas statique ? Si non, quels échanges faut-il effectuer pour obtenir un tas ?
2. Un tableau trié en ordre décroissant représente-t-il un tas ?
3. On veut réaliser une procédure qui purge une pile de toutes les occurrences d'un entier donné (1 dans l'exemple):

- a. Réaliser cette procédure en utilisant une deuxième pile.
- b. Réaliser cette procédure d'une manière récursive sans utiliser aucune autre structure (ni pile ni file)

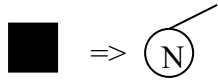


Tournez la page ../..

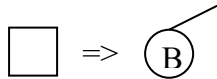
Exercice 3 (6 Pts)

On convient de représenter une image en noire et blanc à l'aide un arbre m-aire de degré 4 comme suit :

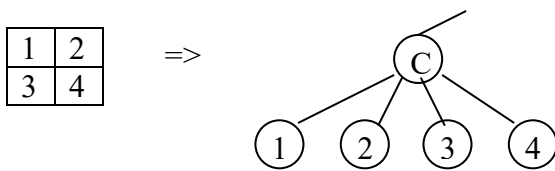
Une image toute noire est représentée par une feuille d'une valeur 'N'



Une image toute blanche est représentée par une feuille d'une valeur 'B'

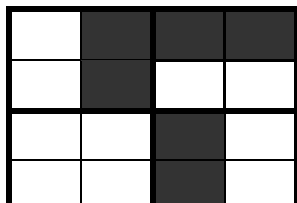


Une image contenant les deux couleurs est représentée par un sous arbre dont la racine est de valeur 'C' et chaque fils représente un quart de l'image.

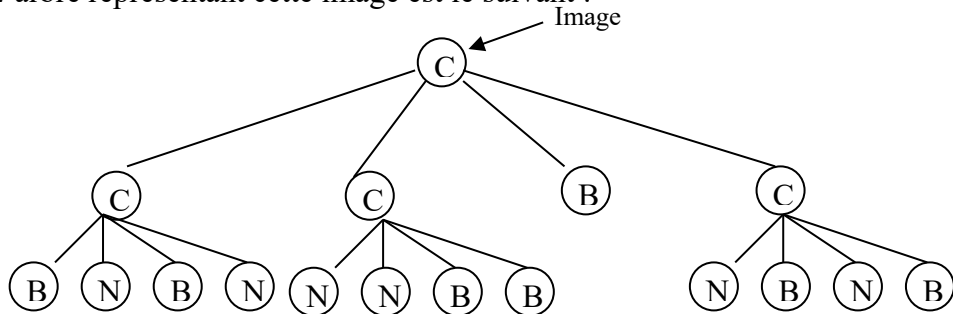


Exemple :

Soit l'image suivante :



L'arbre représentant cette image est le suivant :



1. Donner les déclarations des structures de données nécessaires à l'implémentation de ce modèle.

2. Soit la procédure suivante :

Procédure Traiter (Racine : Pointeur(TNœud)) ;

Var P : Pointeur(TNœud) ;

I : entier ;

Debut

Si Racine ≠ Nil et Valeur(Racine) = 'C' alors

P ← Fils₁ (Racine) ;

Aff_Fils₁ (Racine , Fils₃ (Racine)) ;

Aff_Fils₃ (Racine , Fils₄ (Racine)) ;

Aff_Fils₄ (Racine , Fils₂ (Racine)) ;

Aff_Fils₂ (Racine , P) ;

Pour I=1 à 4 faire

Traiter (Fils_i (Racine)) ;

FPour ;

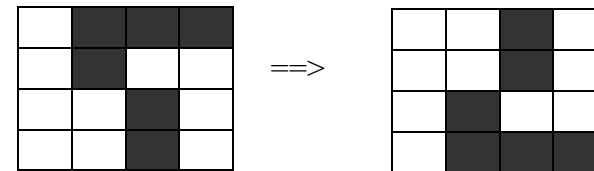
FSi

Fin ;

Donner l'image correspondant à l'arbre précédent après l'appel

Traiter (Image) ;

3. Ecrire la procédure permettant de tourner verticalement une image représentée par un arbre.



Bon courage

A.Djeffal

Corrigé type

Exercice 1

1. (1.5 pt)

Type TLivre = Structure

Titre, Auteur : chaîne ;

Suivant : Pointeur (TLivre)

Fin ;

Type TCategory = Structure

NomCateg : chaîne ;

NbLivre : entier ;

TeteListeLivres : Pointeur (TLivre)

Suivant : Pointeur (TCategory) ;

Fin ;

Var

Bibliotheque : Pointeur (TCategory) ;

Procedure Initialisation ;

Debut

Bibliotheque ← Nil ;

Fin ;

2. (1.5 pt)

Procedure AjouterCategorie (NCateg : chaîne) ;

Var

P, PP, Q : Pointeur (TCategory) ;

Debut

P ← Bibliotheque ;

PP ← Nil ;

TQ P ≠ Nil et NomCteg(P) ≠ NCateg Faire

PP ← P ;

P ← Suivant (P) ;

FTQ ;

Si P ≠ Nil alors Ecrire ('Cette categorie existe déjà') ;

Sinon

Allouer (Q) ;

Aff_Val (Q, NCateg, 0, Nil) ;

Aff_Adr (Q, Nil) ;

Si PP = Nil alors Bibliotheque ← P

Sinon

Aff_Adr (PP, Q) ;

FSi ;

Fin;

3. (2 pts)

Procedure AjouterLivre(NCateg, NTitre, NAuteur:chaîne) ;

Var

P : Pointeur (TCategory) ;

Q : Pointeur (TLivre) ;

Debut

P ← Bibliotheque ;

TQ P ≠ Nil et NomCateg(P) ≠ NCateg Faire

P ← Suivant (P) ;

FTQ ;

Si P = Nil alors Ecrire('Cette catégorie n'existe pas')

Sinon

Allouer (Q) ;

Aff_Adr(Q, TeteListeLivre(P)) ;

Aff_Val(Q, NTitre, NAuteur) ;

Aff_Val(P, NomCateg (P), NbLivre(P) + 1, Q) ;

FSi

Fin ;

4. (1.5 pt)

Procédure Afficher (NCateg : chaîne) ;

Var

P : Pointeur (TCategorie) ;

Q : Pointeur (TLivre) ;

Debut

P ← Bibliothèque ;

TQ P ≠ Nil et NomCateg(P) ≠ NCateg Faire

P ← Suivant (P) ;

FTQ ;

Si P = Nil alors Ecrire('Cette catégorie n'existe pas')

Sinon

Q ← TeteListeLibre(P) ;

TQ Q ≠ Nil Faire

Ecrire (Titre(Q), Auteur(Q)) ;

Q ← Suivant (Q) ;

FTQ ;

FSi

Fin ;

5. (1 pt)

Fonction NbreLivre :Entier ;

Var

P : Pointeur (TCategorie) ;

Nb : entier ;

Debut

Nb ← 0 ;

P ← Bibliothèque ;

TQ P ≠ Nil Faire

Nb ← Nb + NbreLivres(P) ;

P ← Suivant (P) ;

Fin ;

NbreLivre ← Nb ;

Fin ;

6. (2.5 pts)

Procédure SupprimerCategorie (NCateg : chaîne) ;

Var

Debut

P ← Bibliothèque ;

PP ← Nil ;

TQ P ≠ Nil et NomCateg(P) ≠ NCateg Faire

Pp ← Pil.

P ← Suivant (P) ;

FTQ ;

Si P = Nil alors Ecrire('Cette catégorie n'existe pas')

Sinon

Q ← TeteListeLivre(P) ;

TQ Q ≠ Nil Faire

Q1 ← Q ;

Q ← Suivant(Q) ;

Liberer (Q1) ;

FTQ ;

Aff_Adr(PP, Suivant(P)) ;

Liberer (P) ;

FSi

Fin ;

Exercice 2**1. (1 pt)**

Le tableau (1,6,1,6,7,9,1,10,6) n'est pas un tas ; pour obtenir un tas, il faut le mettre dans l'ordre suivant : (10,7,9,6,6,6,1,1,1)

Le tableau (1,3,1,4,3,2,1,4,2) n'est pas un tas ; pour obtenir un tas, il faut le mettre dans l'ordre suivant : (4, 4,2,3,3,1,1,2)

2. (0.5 pt)

Oui un tableau trié dans l'ordre décroissant représente un tas.

3.a. (1.5 pt)

```
Procédure Purger ( P : Pile, n : entier ) ;
```

```
Var
```

```
  P1 : Pile ;
```

```
  x : entier ;
```

```
Debut
```

```
  InitPile(P1) ;
```

```
  TQ non Pile_Vide (P) Faire
```

```
    Depiler ( P, x ) ;
```

```
    Si x ≠ n alors Empiler ( P1 , x ) ;
```

```
  FTQ ;
```

```
  TQ non Pile_Vide(P1) Faire
```

```
    Depiler ( P1 , x ) ;
```

```
    Empiler ( P, x ) ;
```

```
  FTQ ;
```

```
Fin ;
```

b. (2 pts)

```
Procédure Purger( P : Pile, n : entier ) ;
```

```
Var
```

```
  x : entier ;
```

```
Debut
```

```
  Si non Pile_Vide(P) alors
```

```
    Depiler ( P, x ) ;
```

```
    Purger ( P, n ) ;
```

```
    Si x ≠ n alors Empiler ( P, x ) ;
```

```
  FSi
```

```
Fin ;
```

Exercice 3**1. (1 pt)**

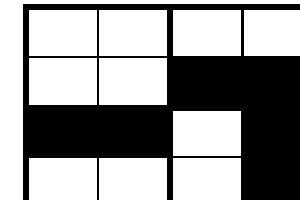
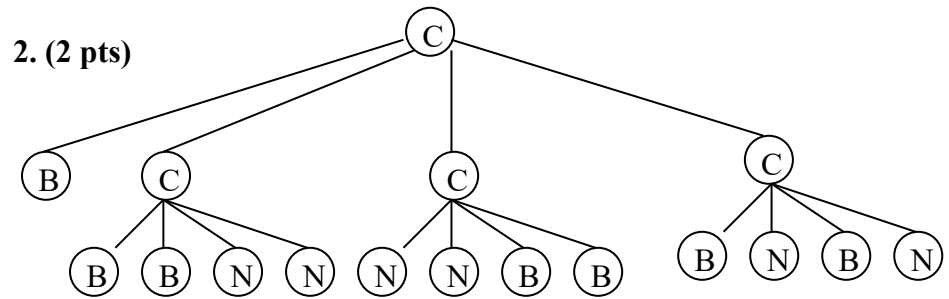
```
Type TNoeud = structure
```

```
  Couleur : caractère ;
```

```
  Fils : Tableau [1..4] de Pointeur (TNoeud) ;
```

```
Fin ;
```

```
Var Image : Pointeur (TNoeud) ;
```

2. (2 pts)

L'image résultante est une rotation de 90° au sens des aiguilles d'une montre.

3. (3 pts)

```
Procédure Traiter (Racine : Pointeur(TNoeud)) ;
```

```
  Var P : Pointeur(TNoeud) ;
```

```
  I : entier ;
```

```
Debut
```

```
  Si Racine ≠ Nil et Valeur(Racine) = 'C' alors
```

```
    P ← Fils1 ( Racine ) ;
```

```
    Aff_Fils1 ( Racine , Fils3 ( Racine ) ) ;
```

```
    Aff_Fils3 ( Racine , P ) ;
```

```
    P ← Fils2 ( Racine ) ;
```

```
    Aff_Fils2 ( Racine , Fils4 ( Racine ) ) ;
```

```
    Aff_Fils4 ( Racine , P ) ;
```

```
  Pour I=1 à 4 faire
```

```
    Traiter ( Filsi ( Racine ) ) ;
```

```
  FPour ;
```

```
  FSi
```

```
Fin ;
```