Structures de données avancées :

Principales structures de données

Pr ZEGOUR DJAMEL EDDINE
Ecole Supérieure d'Informatique (ESI)
www.zegour.uuuq.com
email: d_zegour@esi.dz

Les principales structures de données en RAM

Les arbres binaires

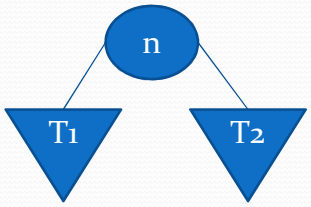
- ***Définition***
  - **Structure de données hiérarchique, généralement dynamique.**
  - **Liste chaînée non linéaire de maillons**
  - **Au maximum 2 fils pour chaque nœud**
  - **Un arbre est défini par son nœud racine.**

- ***Modèle (Machine abstraite)***
  - **Allocation/libération: Créernoeud(x),Libérer(p)**
  - **Accès : Info(p), Fg(p), Fd(p)**
  - **Mise à jour : Aff-info(p, x), Aff-fg(p, x), Aff-fd(p, x)**

Les principales structures de données en RAM

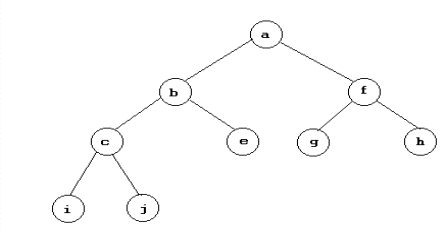
Parcours des arbres binaires

- ***Les plus utilisés***
  - **Préordre : n T1 T2**
  - **Inordre : T1 n T2**
  - **Postordre : T1 T2 n**
- ***Autres***
  - **Préordre inverse : n T2 T1**
  - **Inordre : T2 n T1**
  - **Postordre : T2 T1 n**
  - **Parcours en largeur**
  - **Etc.**



Les principales structures de données en RAM

Parcours des arbres binaires



- **Préordre :**
  a, b, c, i, j, e, f, g, h.
- **Inordre :**
  i, c, j, b, e, a, g, f, h
- **Postordre:**
  i, j, c, e, b, g, h, f, a.

Les principales structures de données en RAM

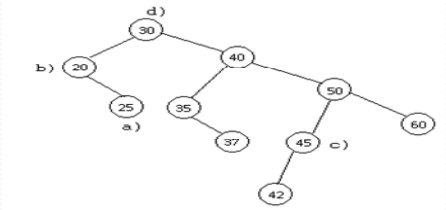
Les arbres de recherche binaire

- **Définition**
  - **Éléments ordonnés par une relation d'ordre**
  - **(Éléments à gauche de x) < x ; (Éléments à droite de x) > x**
  - **L'inordre donne la liste ordonnée de tous les éléments.**

- **Opérations**
  - **Recherche : A chaque étape, soit l'élément recherché est trouvé, soit un sous arbre est éliminé. (recherche dichotomique)**
  - **Insertion: L'élément inséré est toujours une feuille**
  - **Suppression : plus délicate.**

Les principales structures de données en RAM

Les arbres de recherche binaire / Suppression



n (pointeur de l'élément à supprimer)

	fg	fd	Action
a)	^	^	Remplacer n par ^
b)	^	#^	Remplacer n par Fd(n)
c)	#^	^	Remplacer n par Fg(n)
d)	#^	#^	1. Rechercher le plus petit descendant du sous-arbre droit, soit p. 2. Remplacer Info(n) par Info(p) 3. Remplacer p par Fd(p)

Les principales structures de données en RAM

Les arbres m-aires

- **Généralisation de l'arbre binaire**

Chaque nœud peut avoir entre o et n fils

- ***Modèle (Machine abstraite)***

  Allouer(P), Libérer(P)
  Fils (P,I), Aff_fils(P, I, J)
  Info(P), Aff_info(P, Info)
  Nbr(P), Aff_nbr(P, N)

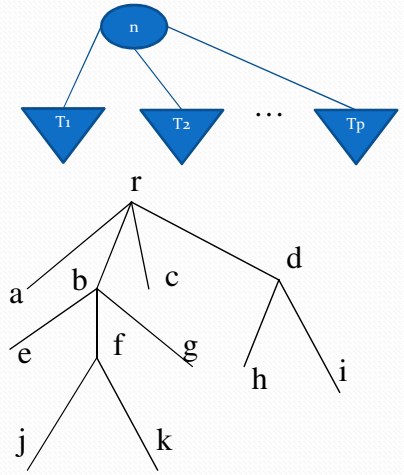
Les principales structures de données en RAM

Parcours des arbres m-aires

Préordre : n T1 T2 .. . Tp
(r a b e f j k g c d h i)

Inordre : T1 n T2 ....Tp
(a r e b j f k g c h d i)

Postordre : T1 T2 ... Tp n
(a e j k f g b c h i d r)



Les principales structures de données en RAM

Les arbres m-aires

- **Transformation arbre m-aire ---> arbre binaire**

❑ **Première étape :**

Lier les frères dans une liste linéaire chaînée

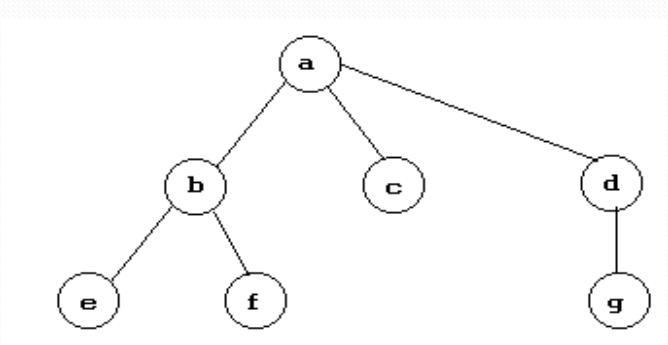
❑ **Deuxième étape**

Rotation de 45 ° dans le sens des aiguilles d'une montre

Les principales structures de données en RAM

Les arbres m-aires

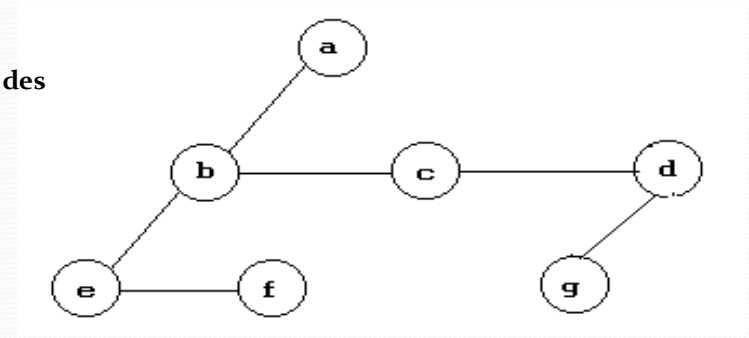
Exemple d'un arbre m-aire



Les principales structures de données en RAM

Les arbres m-aires

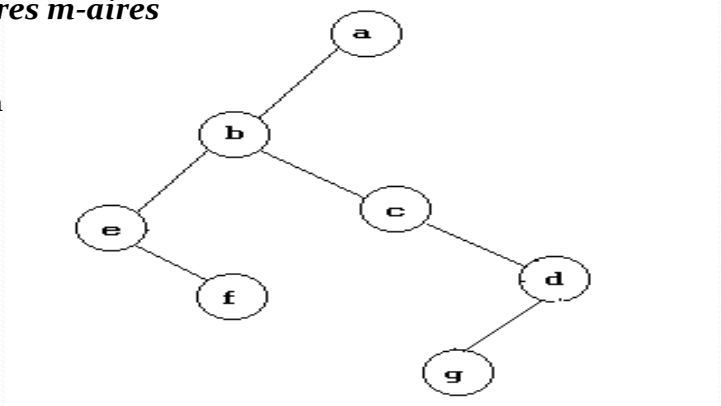
Liaison des frères



Les principales structures de données en RAM

Les arbres m-aires

Rotation



Les principales structures de données en RAM

Arbre de recherche m-aire (ARM)

- **Généralisation de l'ARB**
- **Structure d'un nœud :**

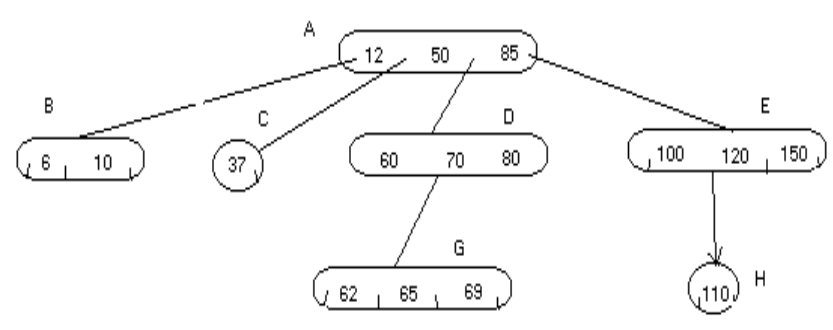
$$(s_1, k_1, s_2, \ldots\ldots k_n\text{-}1, s_n)$$

K_i des données telles que $k_1 < k_2 \ldots < k_{n-1}$

(Éléments dans s_1) <= k_1
(Éléments dans s_j) > k_{j-1} et <= k_j (j=2,3, ...n-1)
(Éléments dans s_n) > k_{n-1}.

Les principales structures de données en RAM

Exemple d'ARM d'ordre 4



Les principales structures de données en RAM

Les arbres de recherche m-aires

- ***Modèle (Machine abstraite)***

  Allouer(P), Libérer(P)
  Fils (P,I), Aff_fils(P, I, J)
  Info(P, I), Aff_info(P, I, Info)
  Nbr(P), Aff_nbr(P, N)

Les principales structures de données en RAM

Arbre de recherche m-aire (ARM)

- ***Arbre de recherche TOP-DOWN : Tout nœud non rempli doit être une feuille.***
  C' est le plus utilisé.

Opérations

- **Recherche : Comme une généralisation de l'arbre de recherche Binaire : O($\log_{Ordre}(n)$)**

Ordre = Ordre de l'ARM.

Les principales structures de données en RAM

Arbre de recherche m-aire (ARM)

Opérations

- **Insertion :**

1. Rechercher la clé.

2. Si clé non trouvée(on est sur une feuille, soit P) et P non complet (c'est à dire Nbr(N) < Ordre-1) on l'insère.

3. Si clé non trouvée et P complet :

- allocation d'un nouveau nœud

- insérer la donnée dans ce nœud

- placer le nœud comme fils du nœud surchargé

Les principales structures de données en RAM

Arbre de recherche m-aire (ARM)

Suppression

- **logique :**

Laisser la clé au niveau du nœud et le marquer

Le nœud reste utilisé pour l'algorithme de recherche

- **Physique (plus chère)**

Technique similaire à celle des arbres de recherche binaire :

Les principales structures de données en RAM

Arbre de recherche m-aire (ARM)

- **Suppression physique**

❑ **1. Si clé à supprimer a un sous arbre gauche ou droit vide**

- **Supprimer la clé**
- **Tasser le nœud.**
- **Si c'est la seule clé dans le nœud, libérer le nœud.**

❑ **2. Si clé à supprimer a des sous arbres gauche et droit tous les deux non vides :**

- **Trouver la clé successeur (qui doit avoir un sous arbre gauche vide).**
- **Remplacer la clé à supprimer par ce successeur**
- **Tasser le nœud qui contenait ce successeur.**
- **Si le successeur est la seule clé dans le nœud, libérer le nœud.**

Les principales structures de données en RAM

Implémentation des arbres de recherche binaire

```
TYPE
  T = ^Noeud;
  Noeud = RECORD
    Element : INTEGER;
    Fg, Fd, Pere  : T ;
  END;
```

PASCAL

```
struct Noeud
  {
    int  Element ;
    struct Noeud *Fg ;
    struct Noeud *Fd ;
  } ;
```

C

Les principales structures de données en RAM

Implémentation des arbres de recherche m-aire

```
TYPE
  T = ^Noeud;
  Noeud = RECORD
    Infor : ARRAY[1..Max] of INTEGER;
    Fils : ARRAY[1..Max] of T;
    Degre : Byte ;
    Pere : T
  END;
```

PASCAL

```
typedef int Typeqq;
 struct Noeud
  {
    Typeqq  Info[Ordre-1] ;
    struct Noeud *Fils[Ordre] ;
    short Degre;
  } ;
```

C

Les principales structures de données en RAM

Rangement des données dans un tableau

- **Rangement séquentiel(O(n))**
- **Rangement ordonné (O(Log(n))**
- ***Rangement aléatoire ou Hachage (Comme une 3ième possibilité avec O(1)***
- ***T*echnique :**

✓ **Transformer la donnée K par une fonction f.**

✓ **f(K) est alors l'emplacement où sera rangée la donnée K.**

- **Le problème : Impossible de trouver une fonction bijective (paradoxe de l'anniversaire)**
- **Existence de techniques de résolution des collisions.**

Les principales structures de données en RAM

Utilisation des techniques de hachage

- **Définir :**
  - ✓ **La fonction de hachage**
  - ✓ **Une méthode dite méthode de résolution des collision (MRC)**
- **Quelques Définitions :**
  - ✓ **Synonymes : données qui ont même adresse par f**
  - ✓ **Adresse primaire : f(donnée) // Adresse secondaire**
  - ✓ **Donnée en débordement : donnée non rangée dans son adresse primaire**

Les principales structures de données en RAM

Les fonctions de hachage

- **Fonction de hachage**
  fonction f telle que o <= f(K) < M qui réduit au maximum le nombre de collisions.
- **Quelques fonctions :**
  - ✓ ***Fonction de division***
    h(K)= K mod M (M premier)
  - ✓ ***Méthode dite du milieu du carré "middle square"***
    Élever au carré et prendre les chiffres du milieu

Les principales structures de données en RAM

Essai linéaire (L)

En cas de collision sur la case k, utilisation de la séquence cyclique

K-1, ……, 0, M-1, M-2, ……K+1

Les principales structures de données en RAM

Essai linéaire : Algorithme de Recherche/Insertion

L1. [Hacher] i := h(K) { 0 <= i < M }
L2. [Comparer]
Si T(i) = K,
l'algorithme se termine avec succès.
Autrement Si T(i) est vide aller à L4.
L3. [Avancer au prochain]
i := i - 1
Si i < 0 : i := i + M Aller à L2.
L4. [Insérer] {recherche est sans succès}
Si N = M - 1
débordement
Sinon
N := N + 1; Marquer T(i) occupé; Donnée(i) := K

La table T est remplie quand N = M - 1.

Les principales structures de données en RAM

Les autres méthodes de hachage

- **Double hachage (D) : séquence cyclique avec un pas déterminée par une autre fonction de hachage**
- **Chaînage interne (I) : chaînage à l'intérieur de la table**
- **Chaînage séparé (S) : Chaînage à l'extérieur de la table**

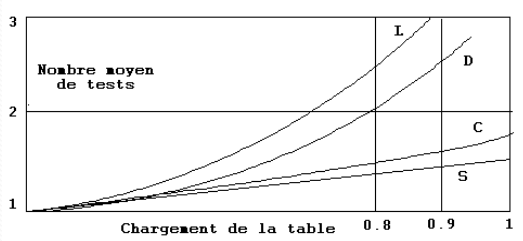
Les principales structures de données en RAM

Classification des méthodes de hachage

- **Première classification**
  - ✓ **Les méthodes ouvertes : L, D, I**
  - ✓ **Les méthodes fermées : S**
- **Deuxième classification**
  - ✓ **Les méthodes avec chaînage : I, S**
  - ✓ **Les méthodes sans chaînage : L, D**

Les principales structures de données en RAM

Performance des méthodes de hachage



S > C > D > L

Les principales structures de données en RAM

Synthèse sur les méthodes de hachage

- **Pas d'ordre**
- **Insertions contrôlées**
- **Données statiques**