

Chap. 2 : Type de données abstrait

Les arbres binaires de recherche

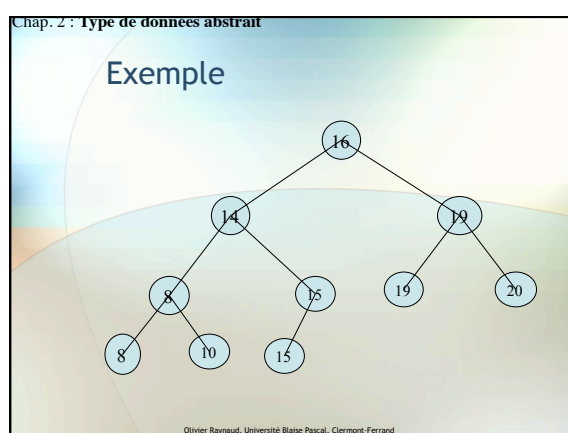
Définition : un arbre binaire est un A.B.R. si pour tout nœud s , les contenus des nœuds du sous-arbre gauche de s sont inférieurs (\leq) au contenu de s et les contenus du sous-arbre droit sont supérieurs (\geq) au contenu de s .

Accesseurs : $a.contenu$; $a.sAG$; $a.sAD$.

Opérations usuelles :

- $a.insérer(x)$: insère l'élément x dans l'arbre;
- $a.maximum()$ et $a.minimum()$
- $a.supprimer(x)$: supprime un élément.
- $a.rechercher(x)$

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand



Chap. 2 : Type de données abstrait

Les « tas »

Définition : un tas est un arbre binaire presque complet tel que pour tous nœuds n sauf la racine on a :
 $n.père.contenu \geq n.contenu$

Accesseurs : $t.contenu$; $t.fG$; $t.fD$; $t.père$

Opérations usuelles :

$t.insérer(x)$: insère l'élément x dans le tas;

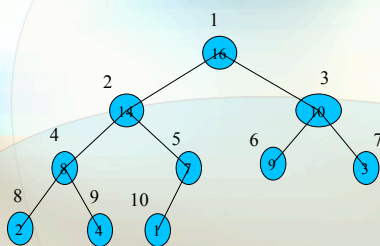
$t.maximum()$: retourne l'élément maximum;

$t.extraire()$: supprime un élément maximum.

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Chap. 2 : Type de données abstrait

Implémentation par un tableau



1	2	3	4	5	6	7	8	9	10
16	14	10	8	7	9	3	2	4	1

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Chap. 2 : Type de données abstrait

Les « tas » : relation de filiation

L'implémentation d'un tas par un tableau admet quelques propriétés :

racine : $nœud\ 1$;

parent du nœud i : $nœud(i \text{ Div } 2)$;

fil gauche du nœud i : $nœud(2i)$;

fil droit du nœud i : $nœud(2i + 1)$

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Définition

Un *type de données abstrait* est composé d'un ensemble d'objets, similaires dans la forme et dans le comportement, et d'un ensemble d'opérations sur ces objets.

L'implémentation d'un T.D.A. ne suis pas de schéma préétabli. Il dépend des objets manipulés et des opérations disponibles pour leur manipulation.

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Contraintes d'implémentation

L'implémentation d'un type de données abstrait doit respecter deux contraintes :

- Utiliser un minimum d'espace mémoire;
- Exécuter un nombre minimal d'instructions pour réaliser une opération.

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

T.D.A. Ensemble dynamique

Définition : On appelle ensemble dynamique *e* un ensemble fini d'éléments issus d'un ensemble discret (entiers, chaîne de caractères,...) et muni d'une relation d'ordre.

Opérations :

e.inserter(x) ajoute un élément *x* à *e*;
e.supprimer(x) un élément *x* de *e*;
e.rechercher(x)
e.maximum() retourne l'élément maximum de *e*;
e.minimum() retourne l'élément minimum de *e*;
e.prédécesseur(x)
e.successeur(x)

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Chap. 2 : Type de données abstrait

T.D.A. Dictionnaire

Définition : On appelle dictionnaire un ensemble dynamique d dont on a restreint l'ensemble des opérations :

Opérations :

$d.insérer(x)$: insère l'élément x dans d ;
 $d.rechercher(x)$: recherche l'élément x dans d ;
 $d.supprimer(x)$: supprime l'élément x de d .

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Chap. 2 : Type de données abstrait

Dictionnaire : Implémentation

Structure de données	Rechercher	Insérer	Supprimer
Tableau non ordonné	$O(n)$	$O(1)$	$O(1)$
Liste non ordonnée	$O(n)$	$O(1)$	$O(1)$
Tableau ordonné	$O(\log n)$	$O(n)$	$O(n)$
Liste ordonnée	$O(n)$	$O(1)$	$O(1)$
Arbre de recherche	$O(h)$	$O(h)$	$O(h)$
Tas	$O(n)$	$O(h)$	$O(h)$

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Chap. 2 : Type de données abstrait

T.D.A. Pile

Définition : Une pile est un ensemble dynamique tel que la suppression concerne toujours le dernier élément inséré. Une telle structure est aussi appelée LIFO (last-in, first out).

Opérations :

$p.empiler(x)$ insère un élément à l'entrée de la pile;
 $p.dépiler()$ retourne et supprime l'élément en entrée de pile;

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Chap. 2 : Type de données abstrait

Les piles : applications

La pile d'exécution : les appels des méthodes dans l'exécution d'un programme sont gérés par une pile.

Éditeur de texte : une pile est fournie par les éditeurs de texte évolués qui possèdent le couple d'actions « annuler-répéter ».

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Chap. 2 : Type de données abstrait

Les piles : Implémentation

On peut implémenter une pile par un couple composé d'un tableau et d'un entier.

1	4	12	8	9	14	20	5	6	2	5
---	---	----	---	---	----	----	---	---	---	---

Inconvénient majeur :
il faut fixer à l'avance la taille maximale de la pile.

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Chap. 2 : Type de données abstrait

T.D.A. File

Définition : Une file est un ensemble dynamique tel que les insertions se font d'un côté (l'entrée de file) et les suppressions de l'autre côté (la sortie de file). Une telle structure est aussi appelé FIFO (first-in, first out).

Opérations :

f.enfiler(x) ajoute un élément en entrée de file;
f.défiler() supprime l'élément situé en sortie de file.

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Chap. 2 : Type de données abstrait

Les files : Implémentation

On peut implémenter une file par un triplet composé d'un tableau et de deux entiers.

entrée

sortie

Inconvénient majeur : il faut fixer à l'avance la taille maximale de la file.

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Chap. 2 : Type de données abstrait

Les files : Implémentation

On peut implémenter une pile par un couple de listes chaînées.

début

fin

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Chap. 2 : Type de données abstrait

Comparaison d'implémentation

Nous avons vu que l'implémentation par les tableaux impose de définir par avance la taille de la file. Ce qui n'est pas cas avec les listes chaînées.

Quelque soit le choix d'implémentation, ce choix n'apparaît pas pour le programmeur puisqu'il n'aura accès à ce type de données que par l'intermédiaire d'un ensemble de méthodes. **La file devient alors un type de données abstrait.**

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

T.D.A. Files de priorité

Définition : Une file de priorité est une structure de données permettant de gérer un ensemble f d'éléments, chacun ayant une priorité associée appelée *clé*.

Opérations :

$f.\text{insérer}(x, \text{clé})$: insère l'élément x dans f ;
 $f.\text{maximum}()$: retourne l'élément de plus grande clé;
 $f.\text{extraireMax}()$: retourne et supprime l'élément de f de plus grande clé.

[CLR90]

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

File de priorité : Implémentation

Structure de données	Insérer()	Maximum()	extraireMax()
Tableau non ordonné	$O(1)$	$O(n)$	$O(n)$
Liste non ordonnée	$O(1)$	$O(n)$	$O(n)$
Tableau ordonné	$O(n)$	$O(1)$	$O(1)$
Liste ordonnée	$O(n)$	$O(1)$	$O(1)$
Tas	à étudier	à étudier	à étudier

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

T.D.A. Famille d'ensembles

Définition : Soit X un ensemble muni d'une relation d'ordre $<_x$, on appelle collection (ou famille) un ensemble F de sous-ensembles de X .

Opérations :

$c.\text{insérer}(s)$: insère le sous-ensemble s dans c ;
 $c.\text{appartient}(s)$: vérifie si le sous-ensemble s est dans c ;
 $c.\text{supprimer}(s)$: supprime le sous-ensemble s de c .

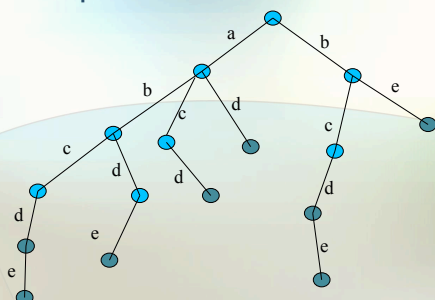
Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Implémentation et complexité

Question : Quelle structure de données permettrait de proposer des algorithmes pour les opérations *d'insertion, de vérification d'appartenance et de suppression* admettant une complexité indépendante de la taille de la famille?

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Exemple



Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

L'arbre lexicographique

Définition : soit F une famille de sous-ensembles de X , nous associons à F un arbre $T(F)$ lexicographique unique tel que :

- chaque arête de l'arbre est étiquetée par un élément de X ;
- à chaque nœud notifié de l'arbre correspond un mot de F ;
- à chaque mot de F correspond un chemin unique dans l'arbre tel que ce mot corresponde à la concaténation des étiquettes de ce chemin;
- l'ordre des arêtes d'un chemin coïncident avec l'ordre $<_X$;
- l'ordre des arêtes sortant d'un nœud coïncident avec l'ordre $<_X$.

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Chap. 2 : Type de données abstrait

Implémentation

Remarque : une représentation d'une collection par un arbre lexicographique correspond à un mapping!

Java Key Mapping :

new() operator : crée un objet de type map et retourne un mapping vide;
get(e) operator : retourne la valeur associée à la clé e si cette clé existe, nil dans le cas contraire;
put(e, value) operator : insère la clé e dans le map et lui associe la valeur *value*.

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Chap. 2 : Type de données abstrait

Comparaison d'implémentation

En Java un mapping est implémenté par des tables de Hachage.

Plusieurs implémentations différentes d'un arbre lexicographique peuvent être proposées en fonction de la façon dont l'ensemble des fils sont représenté :

- Par un tableau;
- Par des listes chaînées;

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Chap. 2 : Type de données abstrait

T.D.A. Gestion de partition

Définition : Une partition p d'un ensemble e est un ensemble de parties non vides de e , deux à deux disjointes et dont la réunion est égale à e .

Opérations :

$p.trouverClasse(e)$: retourne la classe de e dans p ;
 $p.union(c_1, c_2)$: fusionne les deux classes c_1 et c_2 dans p ;

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Chap. 2 : Type de données abstrait

Gestion de partition : Implémentation

1	1	2	3	4	1	2	3	3	1	2	3
0	1	2	3	4	5	6	7	8	9	10	11

Partition : $\{ \{0,1,5,9\}, \{2,6,10\}, \{3,7,8,11\}, \{4\} \}$

Question : Quelle est la complexité des opérations de fusion et de recherche?

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Chap. 2 : Type de données abstrait

Gestion de partition : Implémentation

Partition : $\{ \{0,1,5,9\}, \{2,6,10\}, \{3,7,8,11\}, \{4\} \}$

Père :

0	0	2	3	4	0	2	3	7	5	2	3
0	1	2	3	4	5	6	7	8	9	10	11

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Chap. 2 : Type de données abstrait

Gestion de partition : Implémentation

Question : Quelle est la complexité des opérations de fusion et de recherche avec une implémentation par un tableau « Père »?

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand

Pour résumer

- Nous avons défini un T.D.A. comme un ensemble d'objets cohérent muni d'opérations données. Nous avons dit que l'implémentation d'un T.D.A. devait respecter des contraintes d'efficacité (en espace et en temps).
- Nous avons défini les T.D.A. : ensemble dynamique, dictionnaire, pile, file, file de priorité, collection, gestion de partition.
- L'implémentation de chacun de ces T.D.A. repose sur des structures de données évoquées au chapitre précédent : liste, tableau, arbre, tas, arbre binaire de recherche.

Olivier Raynaud, Université Blaise Pascal, Clermont-Ferrand
