

Les arbres (III)

Nour-Eddine Oussous, Éric Wegrzynowski

Licence ST-A, USTL - API2

30 novembre 2009

Plan du cours

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

1 Algorithmes sur les arbres binaires

- Recherche d'un élément

2 Arbres binaires ordonnés

- Définition et propriété
- Recherche dans un arbre ordonné
- Construction d'arbres ordonnés

Spécification

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

Spécification

$$\begin{array}{lll} \text{recherche} : E \times AB(E) & \longrightarrow & \text{Booleen} \\ & e, a & \longmapsto \begin{cases} \text{Vrai} & \text{si } e \in a \\ \text{Faux} & \text{sinon} \end{cases} \end{array}$$

Spécification

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

Spécification

$$\begin{array}{lll} \text{recherche} : E \times AB(E) & \longrightarrow & \text{Booleen} \\ & e, a & \longmapsto \begin{cases} \text{Vrai} & \text{si } e \in a \\ \text{Faux} & \text{sinon} \end{cases} \end{array}$$

Exemples

$$\text{recherche}(1, \langle 1; \langle 3; \Delta; \Delta \rangle; \Delta \rangle) = \text{Vrai}$$

$$\text{recherche}(2, \langle 1; \langle 3; \Delta; \Delta \rangle; \Delta \rangle) = \text{Faux}$$

Implémentation en Pascal

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

Recherche

```
// recherche(e,a) = VRAI si  $e \in a$ 
//                               FAUX sinon
function recherche(e : ELEMENT;
                  a : ARBRE) : BOOLEAN;
begin
    if estArbreVide(a) then
        recherche := false
    else if racine(a)=e then
        recherche := true
    else
        recherche :=  recherche(e,gauche(a))
                      or recherche(e,droit(a));
end {recherche};
```

Coût de la recherche

Les arbres (III)

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

$c(a)$ = nombre de tests de la condition `estArbreVide(a)`.

Coût de la recherche

Les arbres (III)

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

$c(a)$ = nombre de tests de la condition `estArbreVide(a)`.

Conclusion

- meilleur cas : e se trouve à la racine, on fait un seul test
- pire cas : e n'est pas dans l'arbre ou se trouve dans la dernière feuille, $c(a)$ est linéaire en n

Définition

Les arbres (III)

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

On suppose que l'ensemble E des étiquettes des nœuds des arbres est un ensemble totalement ordonné par une relation notée \leq .

Définition

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

On suppose que l'ensemble E des étiquettes des nœuds des arbres est un ensemble totalement ordonné par une relation notée \leq .

Définition

Un arbre $A \in AB(E)$ est ordonné si

1 il est vide, $A = \triangle$

Définition

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

On suppose que l'ensemble E des étiquettes des nœuds des arbres est un ensemble totalement ordonné par une relation notée \leq .

Définition

Un arbre $A \in AB(E)$ est ordonné si

- 1 il est vide, $A = \Delta$
- 2 ou bien, il ne l'est pas, $A = \langle e; g; d \rangle$, et il vérifie les quatre conditions suivantes
 - g est un arbre binaire ordonné;
 - d est un arbre binaire ordonné;

Définition

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

On suppose que l'ensemble E des étiquettes des nœuds des arbres est un ensemble totalement ordonné par une relation notée \leq .

Définition

Un arbre $A \in AB(E)$ est ordonné si

- 1 il est vide, $A = \Delta$
- 2 ou bien, il ne l'est pas, $A = \langle e; g; d \rangle$, et il vérifie les quatre conditions suivantes
 - g est un arbre binaire ordonné ;
 - d est un arbre binaire ordonné ;
 - e est inférieur ou égal à toutes les étiquettes de d ;

Définition

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

On suppose que l'ensemble E des étiquettes des nœuds des arbres est un ensemble totalement ordonné par une relation notée \leq .

Définition

Un arbre $A \in AB(E)$ est ordonné si

- 1 il est vide, $A = \Delta$
- 2 ou bien, il ne l'est pas, $A = \langle e; g; d \rangle$, et il vérifie les quatre conditions suivantes
 - g est un arbre binaire ordonné ;
 - d est un arbre binaire ordonné ;
 - e est inférieur ou égal à toutes les étiquettes de d ;
 - e est supérieur ou égal à toutes les étiquettes de g ;

Définition

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

On suppose que l'ensemble E des étiquettes des nœuds des arbres est un ensemble totalement ordonné par une relation notée \leq .

Définition

Un arbre $A \in AB(E)$ est ordonné si

- 1 il est vide, $A = \Delta$
- 2 ou bien, il ne l'est pas, $A = \langle e; g; d \rangle$, et il vérifie les quatre conditions suivantes
 - g est un arbre binaire ordonné ;
 - d est un arbre binaire ordonné ;
 - e est inférieur ou égal à toutes les étiquettes de d ;
 - e est supérieur ou égal à toutes les étiquettes de g ;

L'ensemble des arbres binaires ordonnés est noté $ABO(E)$.

Exemples

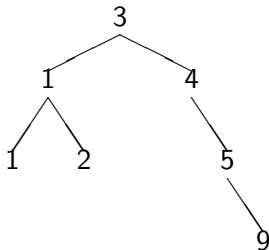
Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés



Exemples

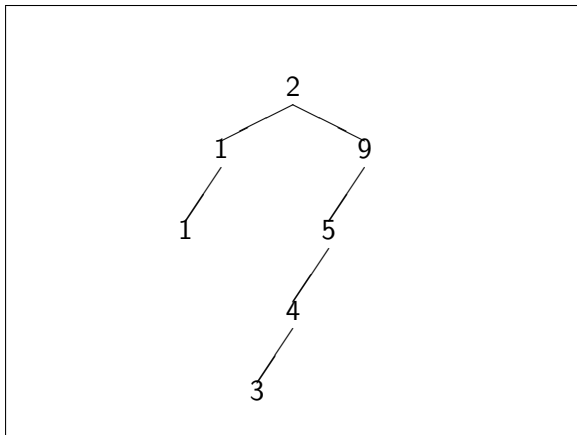
Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés



Propriété caractéristique

Les arbres (III)

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

Théorème

Soit $A \in AB(E)$ un arbre binaire.

A est un arbre binaire ordonné si et seulement si la liste des étiquettes des nœuds dans l'ordre infixé est une liste triée dans l'ordre croissant.

Spécification

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

Spécification

$$\begin{array}{lcl} \text{rechercheABO} : E \times ABO(E) & \longrightarrow & \text{Booleen} \\ & (e, a) \longmapsto & \begin{cases} \text{Vrai} & \text{si } e \in a \\ \text{Faux} & \text{sinon} \end{cases} \end{array}$$

Spécification

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

Spécification

$$\begin{array}{lll} \text{rechercheABO} : & E \times ABO(E) & \longrightarrow \text{Booleen} \\ & e, a & \longmapsto \begin{cases} \text{Vrai} & \text{si } e \in a \\ \text{Faux} & \text{sinon} \end{cases} \end{array}$$

Exemples

$$\begin{aligned} \text{rechercheABO}(1, \langle 3; \langle 1; \langle 1; \Delta; \Delta \rangle; \langle 2; \Delta; \Delta \rangle \rangle \\ \quad \langle 4; \Delta; \langle 5; \Delta; \langle 9; \Delta; \Delta \rangle \rangle \rangle) &= \text{Vrai} \\ \text{rechercheABO}(6, \langle 3; \langle 1; \langle 1; \Delta; \Delta \rangle; \langle 2; \Delta; \Delta \rangle \rangle \\ \quad \langle 4; \Delta; \langle 5; \Delta; \langle 9; \Delta; \Delta \rangle \rangle \rangle) &= \text{Faux} \end{aligned}$$

Implémentation en Pascal

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

Recherche dans ABO

```
// rechercheABO(e,a) = VRAI si  $e \in a$   
//                               FAUX sinon  
function rechercheABO(e : ELEMENT;  
                      a : ARBRE) : BOOLEAN;  
begin  
    if estArbreVide(a) then  
        rechercheABO := false  
    else if racine(a) = e then  
        rechercheABO := true  
    else if racine(a) > e then  
        rechercheABO := rechercheABO(e, gauche(a))  
    else rechercheABO := rechercheABO(e, droit(a));  
end {rechercheABO};
```

Coût de la recherche

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

$c(a)$ = nombre de tests de la condition `estArbreVide(a)`.

- meilleur cas : l'élément se trouve à la racine, un seul test
- pire cas : ce nombre dépend de la hauteur de l'arbre
 - si l'arbre est bien équilibré, pour un ABO de n nœuds, ce nombre sera majoré par $\log_2(n)$
 - si l'arbre est un peigne, ce nombre sera proportionnel à n

Conclusion

Le coût de la fonction `rechercheABO` dépend de la forme de l'arbre.

Spécification

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

Spécification

$$\begin{array}{ccc} \text{insererABO} : E \times ABO(E) & \longrightarrow & ABO(E) \\ e, a & \longmapsto & a' \end{array}$$

a' = arbre a modifié par l'ajout en feuille du nouvel élément e .

Spécification

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

Spécification

$$\begin{array}{ccc} \text{insererABO} : E \times ABO(E) & \longrightarrow & ABO(E) \\ e, a & \longmapsto & a' \end{array}$$

a' = arbre a modifié par l'ajout en feuille du nouvel élément e .

Exemples

$$\begin{array}{lcl} \text{insererABO}(4, \langle 3; \langle 1; \Delta; \Delta \rangle; \Delta \rangle) & = & \langle 3; \langle 1; \Delta; \Delta \rangle; \Delta \rangle \\ \text{insererABO}(1, \langle 3; \langle 1; \Delta; \Delta \rangle; \langle 4; \Delta; \Delta \rangle \rangle) & = & \langle 3; \langle 1; \langle 1; \Delta; \Delta \rangle; \Delta \rangle; \langle 4; \Delta; \Delta \rangle \rangle \end{array}$$

Exemples

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

3141592

Exemples

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

141592
3

Exemples

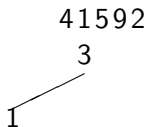
Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés



Exemples

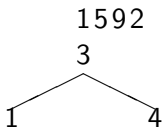
Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés



Exemples

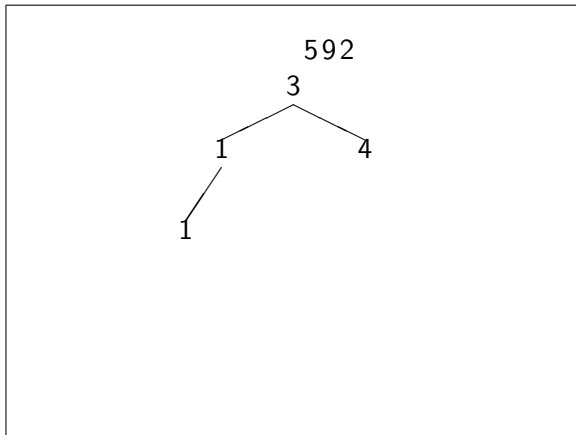
Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés



Exemples

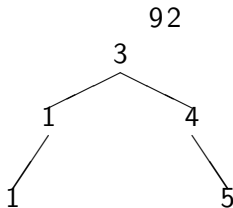
Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés



Exemples

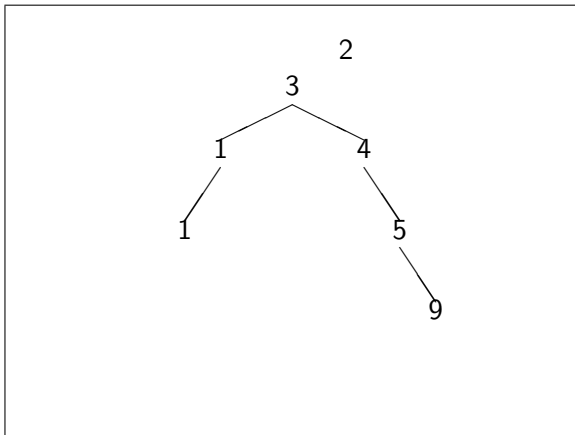
Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés



Exemples

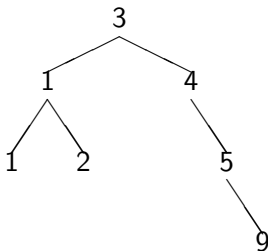
Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés



Exemples

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

3141592

Exemples

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

314159
2

Exemples

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

31415

2

9



Exemples

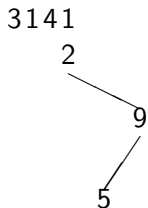
Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés



Exemples

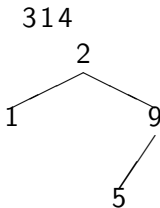
Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés



Exemples

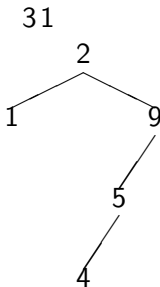
Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés



Exemples

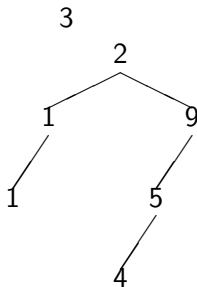
Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés



Exemples

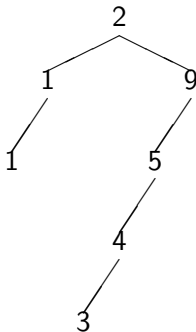
Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés



Implémentation en Pascal

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

Insertion dans ABO

```
// insererABO(e,a)
// ajoute un nouvel élément e en feuille de
// l'arbre a. L'arbre a est modifié
procedure insererABO(e : ELEMENT;
                    var a : ARBRE) ;

var
    b: ARBRE ;

begin
    if estArbreVide(a) then
        a := creerArbre(e, ARBREVIDE, ARBREVIDE)
    else if racine(a) <= e then begin
        b := droit(a) ;
        insererABO(e, b) ;
        modifierDroit(a, b) ;
    end else begin
        b := gauche(a) ;
        insererABO(e, b) ;
        modifierGauche(a, b) ;
    end ;
end {insererABO};
```

Coût de l'insertion

Les arbres (III)

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Algorithmes
sur les arbres
binaires

Arbres binaires
ordonnés

$c(a)$ = nombre de tests `estArbreVide(a)`.

- L'insertion se fait en feuille
- Ce nombre dépend de la hauteur de l'arbre
 - si l'arbre est bien équilibré, pour un ABO de n nœuds, ce nombre sera majoré par $\log_2(n)$
 - si l'arbre est un peigne, ce nombre sera proportionnel à n
- Il est donc souhaitable d'équilibrer les arbres au fur et à mesure des opérations d'insertion

Conclusion

Le coût de la fonction `insérerABO` dépend de la forme de l'arbre.