

- Soit un serveur d'impression où on affecte à chaque requête d'impression un nombre, plus ce numéro est élevé plus le document doit être imprimé rapidement
- Le serveur doit donc faire rapidement les trois opérations suivantes:
 - trouver un maximum
 - retirer cet élément
 - ajouter un nouvel élément

Question: Comment implémenter une telle tâche?

68

- Prendre un tableau, le trier dans l'ordre croissant des priorités. Trouver un maximum et le retirer est alors facile. L'insertion est plus compliquée ...
- Gérer une file simple, rechercher le maximum à chaque fois, le retirer ...

ou

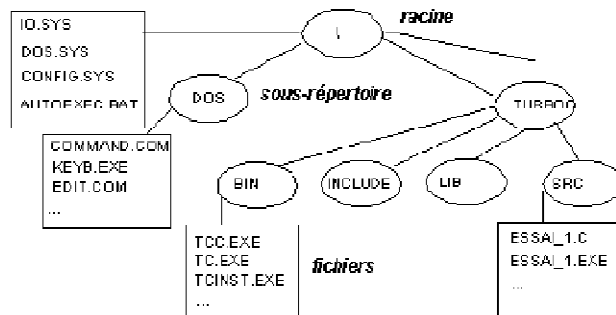
- Considérer une structure à ordre partiel:

un arbre

69

Structures arborescentes

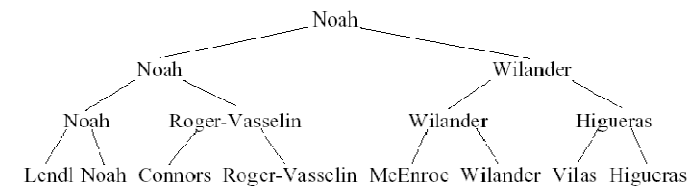
Collections hiérarchisées: Exemples



70

Structures arborescentes

Collections hiérarchisées



71

Structures arborescentes

Quelques définitions:

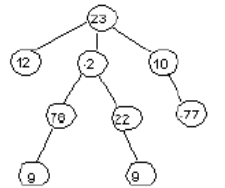
Arbre vide

Arbre singleton

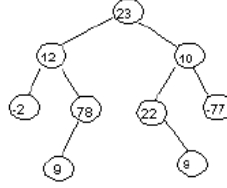
Arbre filiforme



Arbre quelconque



Arbre binaire

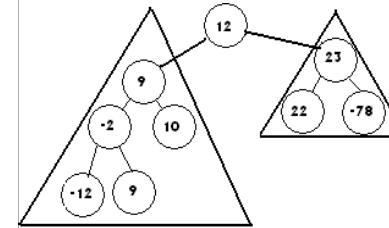


72

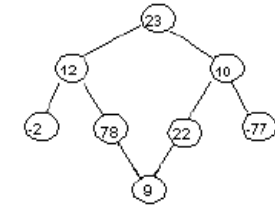
Structures arborescentes

Quelques définitions:

Arbre binaire de recherche



CECI N'EST PAS UN ARBRE!

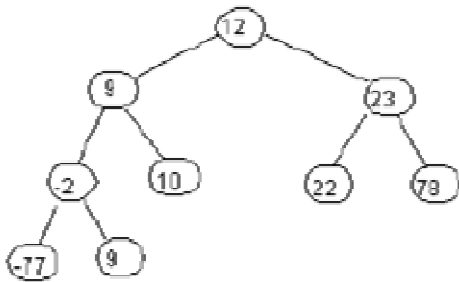


73

Structures arborescentes

Quelques définitions:

Arbre binaire de recherche



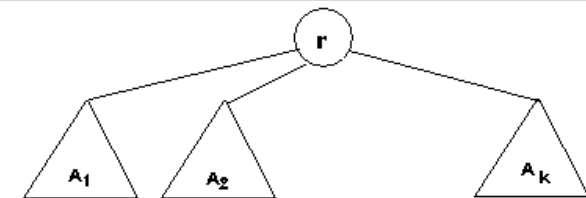
74

Structures arborescentes

Quelques définitions:

- C'est une structure récursive: un arbre est constitué d'un nœud appelé racine et de sous arbres fils de la racine.

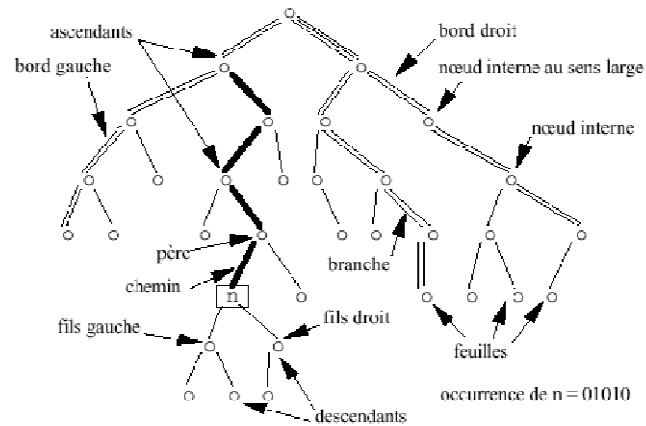
$A = \emptyset$ ou



75

Structures arborescentes

Terminologie



76

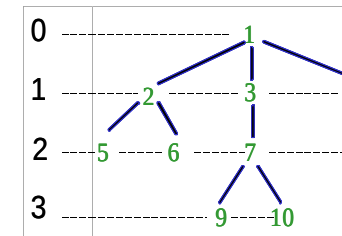
Arbres: Caractéristiques

■ Nombre de nœuds:

$$\begin{aligned} \text{Taille}(A) &= 0 && \text{si } A = \emptyset \\ &1 + \text{taille}(A_1) + \text{taille}(A_2) && \text{si } A = \langle o, A_1, A_2 \rangle \end{aligned}$$

■ Nombre de feuilles d'un arbre:

$$\begin{aligned} \text{Nf}(A) &= 0 && \text{si } A = \emptyset \\ &\max(1, \text{Nf}(A_1) + \text{Nf}(A_2)) && \text{sinon} \end{aligned}$$



77

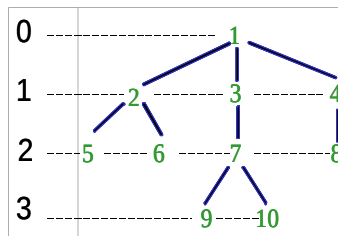
Arbres: Caractéristiques

■ Hauteur d'un nœud: distance de x à la racine

$$\begin{aligned} h(x) &= 0 && \text{si } x = \text{racine}(A) \\ &1 + \text{niveau}(\text{parent}(x)) && \text{sinon} \end{aligned}$$

■ Hauteur d'un arbre:

$$\begin{aligned} H(A) &= -1 && \text{si } A = \emptyset \\ &1 + \max(h(A_1), h(A_2)) && \text{sinon} \end{aligned}$$



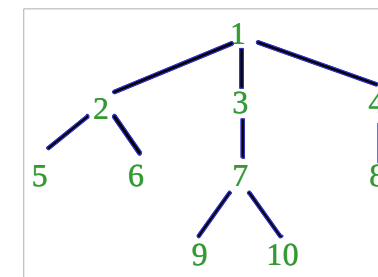
78

Arbres : Parcours en profondeur

$$A = (r, A_1, A_2, \dots, A_k)$$

■ Parcours préfixe

$$\begin{aligned} P(A) &= (r).P(A_1). \dots .P(A_k) \\ &(1, 2, 5, 6, 3, 7, 9, 10, 4, 8) \end{aligned}$$



79

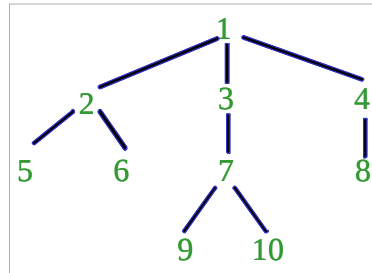
Arbres : Parcours en profondeur

$$A = (r, A_1, A_2, \dots, A_k)$$

■ Parcours suffixe

$$S(A) = S(A_1) \dots S(A_k) \cdot (r)$$

(5, 6, 2, 9, 10, 7, 3, 8, 4, 1)



80

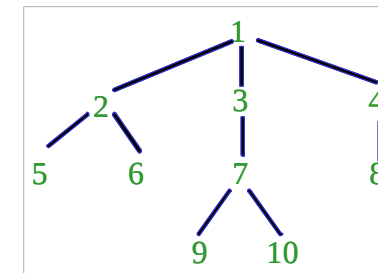
Arbres : Parcours en profondeur

$$A = (r, A_1, A_2, \dots, A_k)$$

■ Parcours symétrique (ou interne ou infixe)

$$I(A) = I(A_1) \cdot (r) \cdot I(A_2) \dots I(A_k)$$

(5, 2, 6, 1, 9, 7, 10, 3, 8, 4)



81

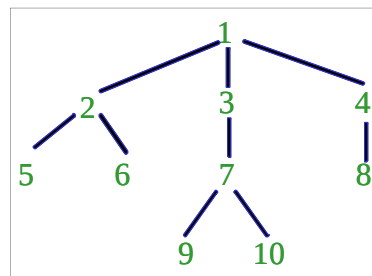
Arbres : Parcours en largeur

$$A = (r, A_1, A_2, \dots, A_k)$$

$$H(A) = (r, x_1, \dots, x_i, x_{i+1}, \dots, x_j, x_{j+1}, \dots, x_n)$$

nœuds de niveau 0, 1, 2, ...

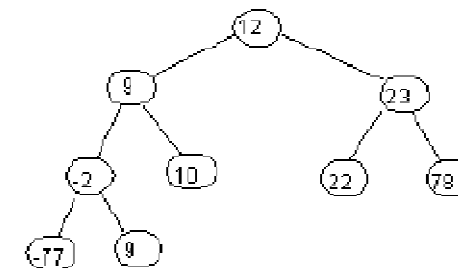
(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)



82

Arbres : Parcours en profondeur

Cas d'un arbre binaire de recherche:



Parcours préfixe : 12, 9, -2, -77, 9, 10, 23, 22, 78

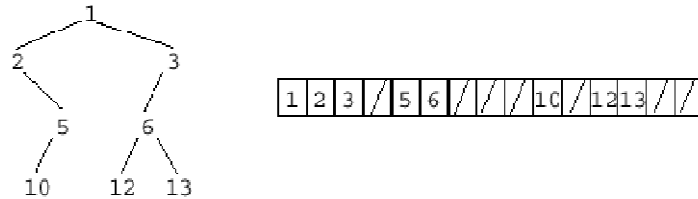
Parcours infixe : -77, -2, 9, 9, 10, 12, 22, 23, 78

Parcours suffixe : -77, 9, -2, 10, 9, 22, 78, 23, 12

83

Arbres: Implémentations

Peut se faire avec des tableaux: ...



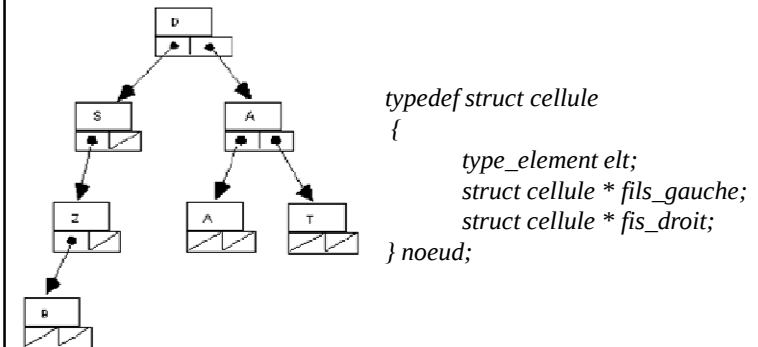
... Pas conseillé !

84

Arbres: Implémentations

Il vaut mieux voir du côté de l'allocation dynamique:

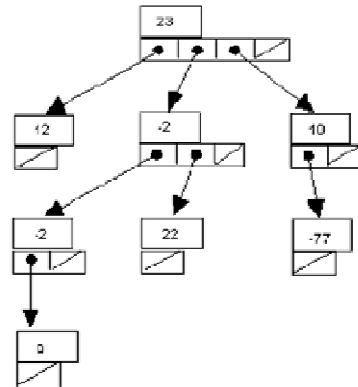
■ Pour les arbres binaires pas de problèmes ...



85

Arbres: Implémentations

■ Pour les arbres généraux, c'est plus délicat!

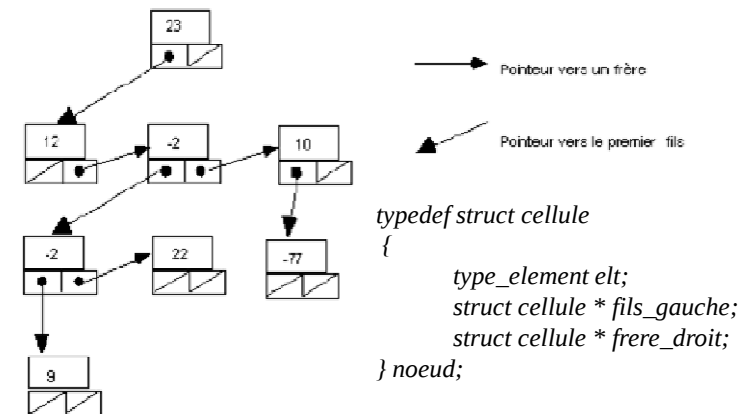


Problème:
Nombre variable de
pointeurs par nœud ...

86

Arbres: Implémentations

■ Mais on peut envisager la structure:

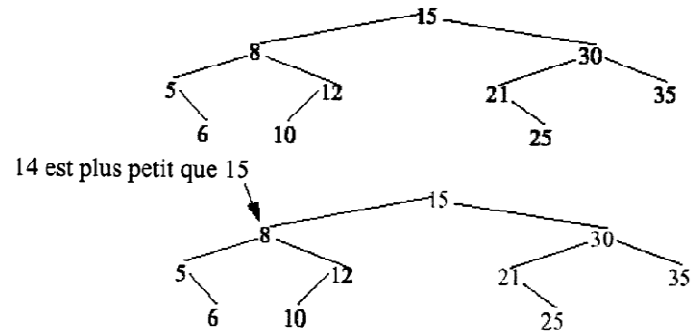


87

Structures arborescentes

Insertion d'une feuille:

Soit à ajouter "14" dans l'arbre binaire de recherche suivant

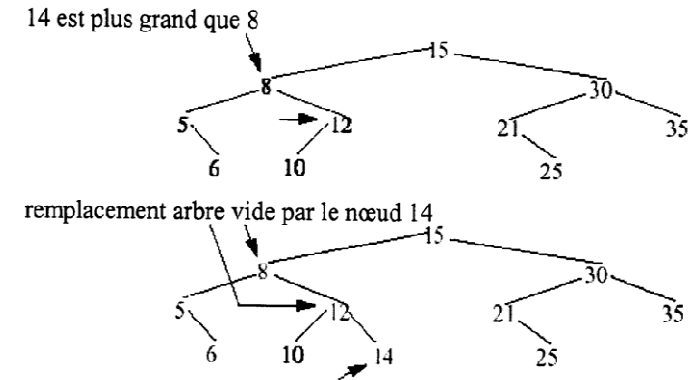


88

Structures arborescentes

Insertion d'une feuille: suite

Soit à ajouter "14" dans l'arbre binaire de recherche suivant

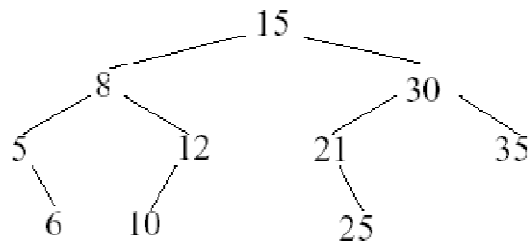


89

Structures arborescentes

adjonction à la racine

Soit à ajouter "11" à l'arbre binaire de recherche suivant en tant que racine.

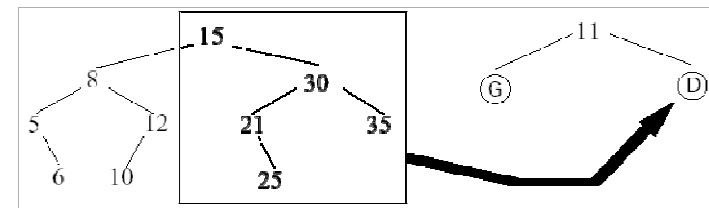


90

Structures arborescentes

adjonction à la racine

- 15 est plus grand que 11 donc à droite de 11
- ainsi que tous les neuds du sous-arbre droit de 15
- ceux qui restent sont plus petits que 15

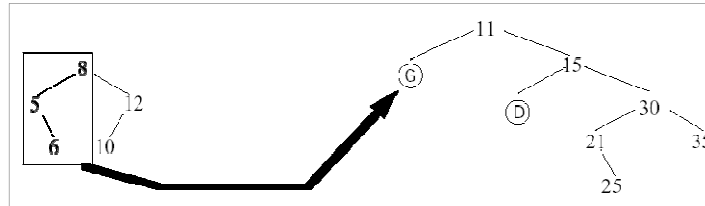


91

Structures arborescentes

adjonction à la racine

- 8 est plus petit que 11
- Ainsi que tous ceux du sous-arbre gauche de 8
- ceux qui restent sont plus petits que 15 et plus grands que 8

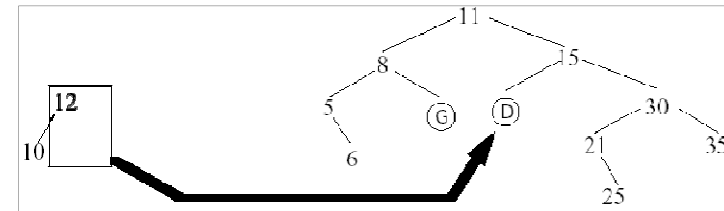


92

Structures arborescentes

adjonction à la racine

- 12 est plus grand que 11 (et plus petit que 15)
- Ainsi que ceux de son sous-arbre droit
- ceux qui restent sont plus grands que 8 et plus petits que 12

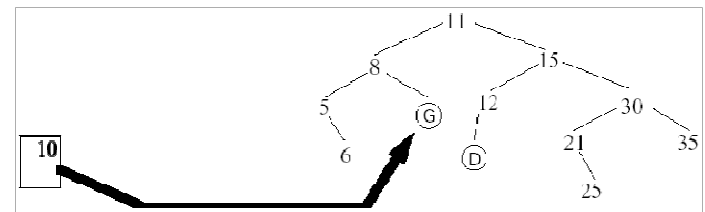


93

Structures arborescentes

adjonction à la racine

- 10 est plus petit que 11 (et plus grand que 8)
- Ainsi ceux de son sous-arbre gauche
- ceux qui restent sont plus grands que 10 et plus petits que 12

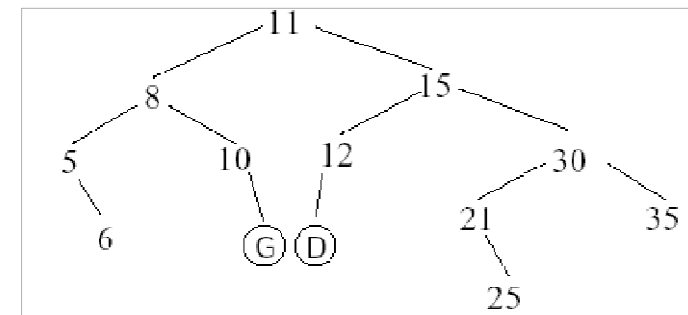


94

Structures arborescentes

adjonction à la racine

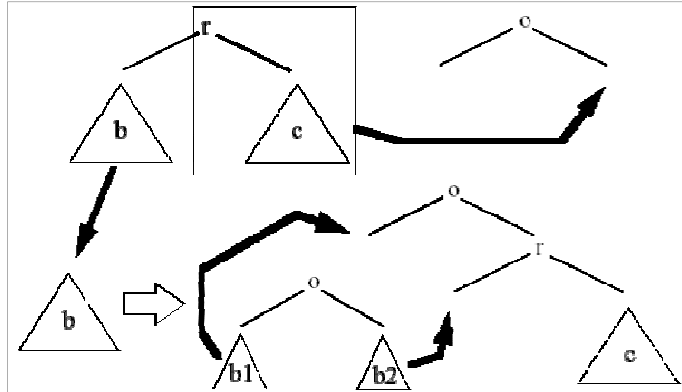
- l'arbre restant est vide: les arbres G et D sont vides.



95

Structures arborescentes

adjonction à la racine: Généralisation (cas $o < r$)



$$\text{ajout}(o, \langle r, b, c \rangle) = \langle o, g(\text{ajout}(o, b), \langle r, d(\text{ajout}(o, b)), c \rangle) \rangle$$

96

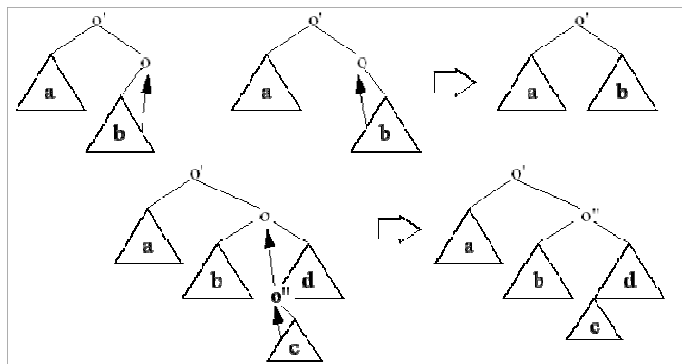
Structures arborescentes

- adjonction en feuille:
les nœuds les plus anciens sont proches de la racine, et sont retrouvés plus vite.
- adjonction en racine:
les nœuds les plus récents sont proches de la racine, et sont retrouvés plus vite.

97

Structures arborescentes

Suppression
rechercher le nœud, puis le supprimer: 3 configurations:



98

Structures arborescentes

- La complexité des ajouts et des suppressions dépend de la hauteur de l'arbre.
- Les ajouts et les suppressions aléatoires de nœuds peuvent aboutir à des hauteurs des nœuds très différentes entre les branches.
- Si on refuse les cas défavorables on peut rééquilibrer les arbres ...

99

Structures arborescentes

Arbres H-équilibrés

A étant un arbre binaire; on définit:

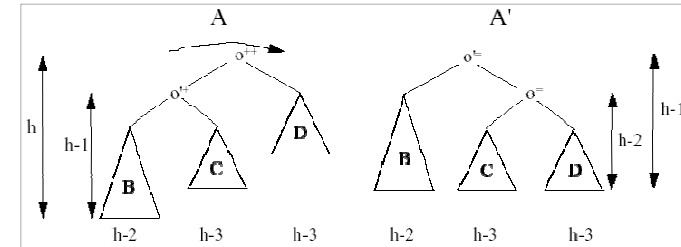
$$\text{Déséquilibre}(A) = \begin{cases} 0 & \text{si } A = \emptyset \\ h(g(A)) - h(d(A)) & \text{sinon} \end{cases}$$

A est un arbre H-équilibré si:
pour tout b sous arbre de A
 $\text{déséquilibre}(b) \in \{-1, 0, 1\}$

100

Structures arborescentes

Exemple de rééquilibrage: Rotation à droite



A arbre binaire de recherche $\Rightarrow$ A' arbre binaire de recherche

Si B, C, D sont H-équilibré $\Rightarrow$ A' est H-équilibré

101