

Arbres en algorithmique (1)

Gilles Aldon, Jérôme Germoni, Jean-Manuel Mény

IREM de Lyon

Mars 2012

Omniprésence des arbres en informatique

Structure ubiquitaire, qualifiée par Donald Knuth de *most fundamental structure in computer science* (*The art of computer programming*).

Idée-clé

Façon d'organiser les informations de sorte à y accéder rapidement.



Définition

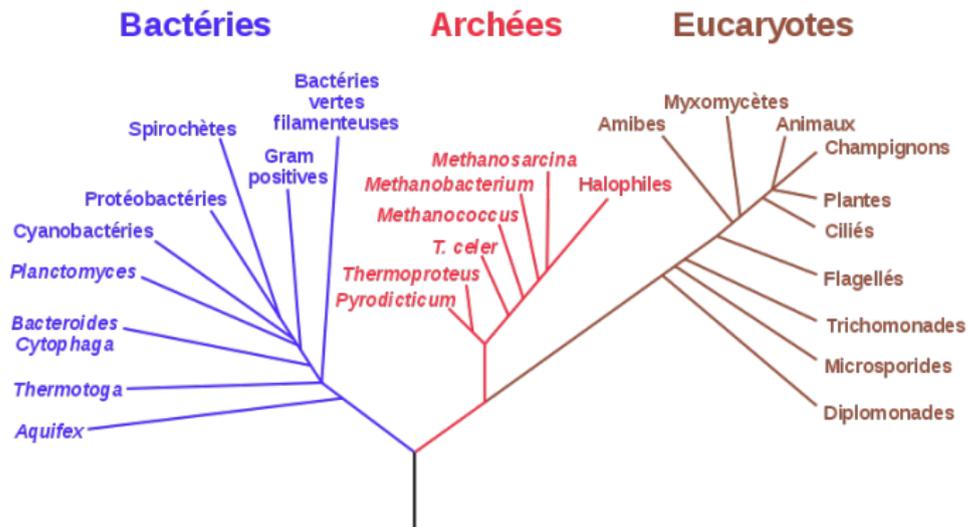
Arbre : graphe sans cycle avec un sommet marqué (*racine*).

Remarque : éventuellement des structures supplémentaires (étiquettes associées aux sommets, marquage de certaines arêtes...).

Quelques exemples d'arbres

Situations \pm informatiques où la structure d'arbre apparaît :

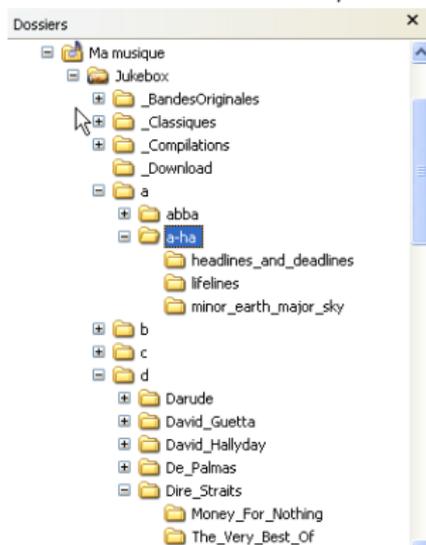
- 1 arbre phylogénétique des espèces (en évolution...),



Quelques exemples d'arbres

Situations \pm informatiques où la structure d'arbre apparaît :

- 1 arbre phylogénétique des espèces (en évolution...),
- 2 arborescence des répertoires d'un ordinateur,



Quelques exemples d'arbres

Situations \pm informatiques où la structure d'arbre apparaît :

- 1 arbre phylogénétique des espèces (en évolution...),
- 2 arborescence des répertoires d'un ordinateur,
- 3 hiérarchie des rubriques d'un site ou fils des réponses dans un forum.

The screenshot shows the website for IREM de Lyon (Institut de recherche sur l'enseignement des mathématiques de Lyon). The page features a navigation menu on the left with a tree-like structure of links. The main content area displays a news item titled "IREM et APMEP : de longues relations".

Navigation Menu (Left):

- Documents stages
- Le coin des formateurs
- L'IREM et le réseau
- Actualités
- Formation
- Groupes de travail
- Documents en ligne
- Archives
- Publications - ventes
- Accès à IREM
- Mots-clés
- Sites favoris
- Sur le Web
- Connexion
- Statistiques
 - Dernière mise à jour: mercredi 8 juin 2011
 - Publication: 350 Articles, Aucun album photo, 114 Brèves

Main Content Area:

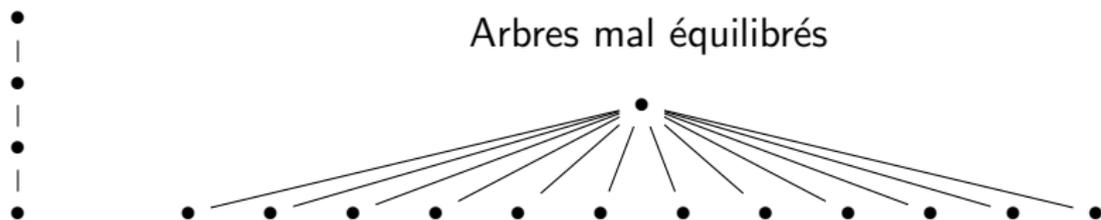
IREM et APMEP : de longues relations

Les IREM ne font partie de l'institution : composante d'une université libre, il ne faut pas les confondre avec l'association loi 1901 qu'elle est. Cependant, les IREM ne seraient pas nés sans le vivier des enseignants de l'Université. De nombreux projets communs unissent les IREM.

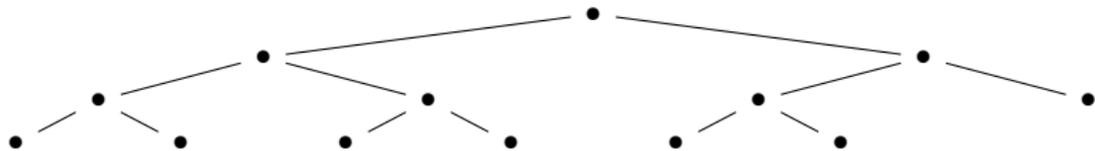
Navigation Menu (Right):

- Astronomie
- Mathématiques, TICE et surdit
- 36 élèves, 36 calculatrices
- Collège
- Lycée
- e-Colab
- Ecole-college
- Démarche d'investigation (et épreuve pratique)
- Environnements numériques de travail
- L'expérimental en mathématiques
- Faites des mathématiques
- Fautes @ problèmes
- Géométrie dynamique
- Lycée professionnel
- Lycée-université
- Projets à Lyon, c'est l'organisation du Rallye Interjets de l'APMEP et le régional de Lyon.
- Calculs de base - Prise en main
- Statistiques et Probabilités
- Fonctions
- Suites numériques
- Algorithmique
- Kit de survie
- Mathématiques
- 10 : Soutenance de thèse Nicolas Pelay : « et didactique et ludique en contexte d'animal
- 14 h 30 : Accès libre à des ateliers interactifs
- 30 : Conférences et ateliers :
- ne : Pourquoi faire simple quand on peut faire
- at : Des webcams pour enseigner et diffuser
- er - Sylvain Gravier : Math-à-modeler (45 mn
- ise
- gratuite, et nous vous serions reconnaissant

Arbres équilibrés



Arbre binaire (presque) complet



Profondeur h , nb sommets : $2^h \leq n < 2^{h+1}$, d'où $h \simeq \log_2 n$.

Condition de parcours rapide d'un arbre

Profondeur **logarithmique** en le nombre de sommets.

Algorithmes avec arbres « cachés »

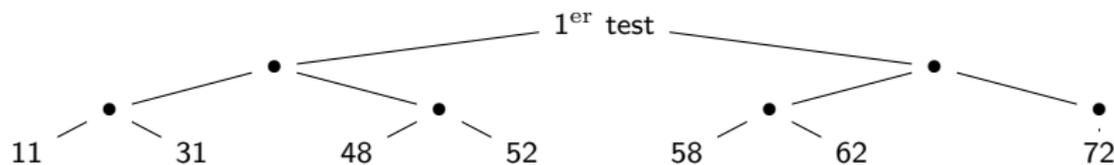
Deux algorithmes où la structure d'arbre reste implicite :

- 1 recherche dichotomique dans une liste triée :
est-ce que 48 figure dans [11, 31, 48, 52, 58, 62, 72] ?

Algorithmes avec arbres « cachés »

Deux algorithmes où la structure d'arbre reste implicite :

- 1 recherche dichotomique dans une liste triée :
est-ce que 48 figure dans [11, 31, 48, 52, 58, 62, 72] ?



Algorithmes avec arbres « cachés »

Deux algorithmes où la structure d'arbre reste implicite :

- 1 recherche dichotomique dans une liste triée :
- 2 problème des reines sur l'échiquier :

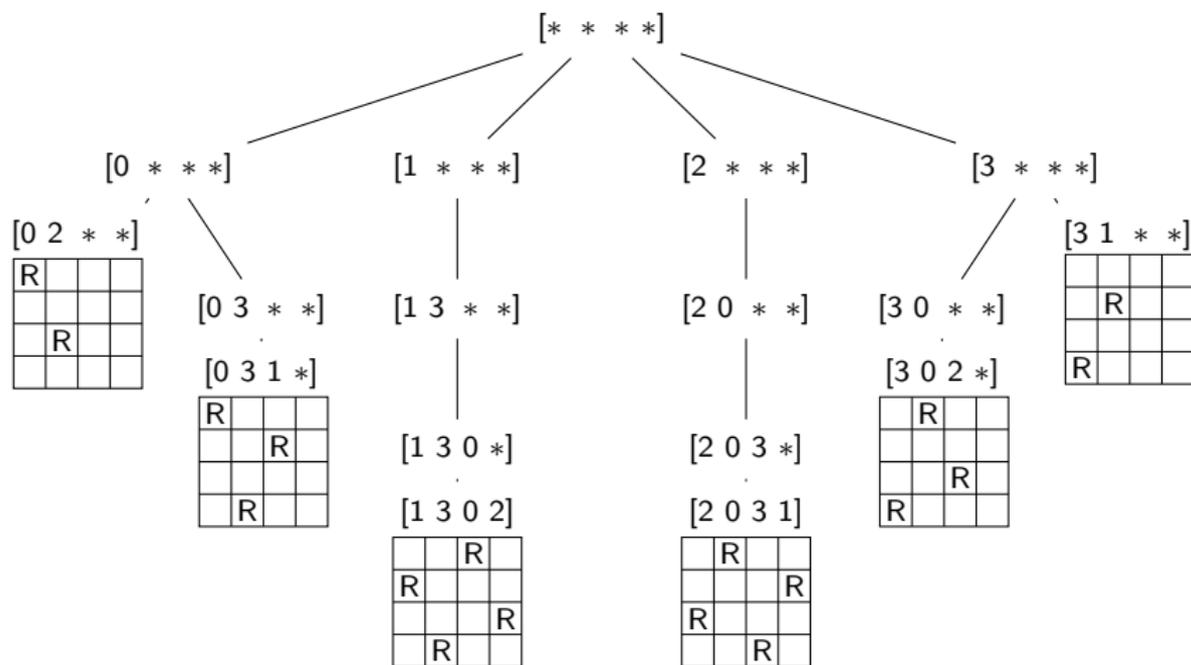
Placer n reines sur un échiquier $n \times n$
sans qu'elles se menacent entre elles.

Algorithme (*backtracking*) :

- placer une reine sur la première colonne à tous les endroits possibles, placer une reine sur la deuxième colonne à tous les endroits encore autorisés, etc. ;
- si ça aboutit, stocker la solution ;
- revenir en arrière : placer la dernière reine posée à l'endroit possible suivant.

Un arbre pour les reines

Interprétation : parcours « main gauche » de l'arbre suivant :



Un arbre pour les fougères

Tracé d'attracteurs pour les *iterated function systems* (IFS)

Algorithmes utilisant des arbres

- algorithme de Huffman
 - compression de données sans perte (JPEG) :
 - entrée : un texte à coder
 - sortie : une suite de 0 et 1
 - principe : consiste à construire un arbre binaire, à mettre les lettres sur les feuilles et à les repérer par le chemin sur l'arbre ;
 - algorithme de Dijkstra :
 - entrée : un graphe dont les arêtes portent des poids ≥ 0 , un sommet r ,
 - sortie : le plus court chemin de r vers tous les autres sommets,
 - principe : parcours en largeur du graphe, principe glouton
- Réinterprétation : produit un **arbre couvrant** du graphe (réunion des plus courts chemins issus de r).
- tri par tas (*heapsort*) [exemple : gif animé]

Algorithme de Huffman

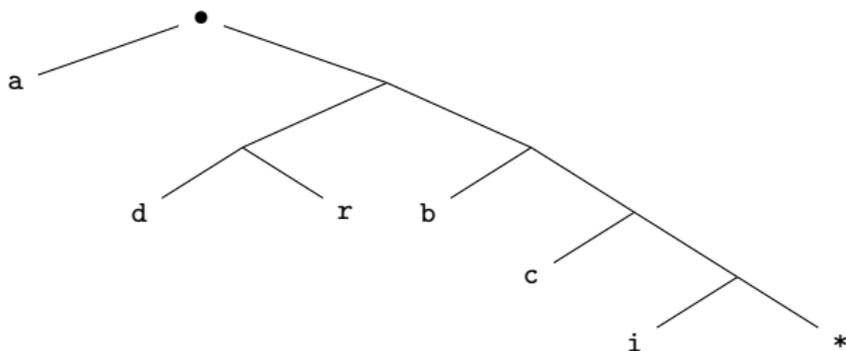
Entrée : abracadabri*abracadabra

Lettres	a	b	c	d	i	r	*	coût
Nombre	9	4	2	2	1	4	1	23
Codage naïf	000	001	010	011	100	101	110	69

Algorithme de Huffman

Entrée : abracadabri*abracadabra

Sortie :



Lettres	a	b	c	d	i	r	*	coût
Nombre	9	4	2	2	1	4	1	23
Codage naïf	000	001	010	011	100	101	110	69
Huffman	0	110	1110	100	11110	101	11111	57

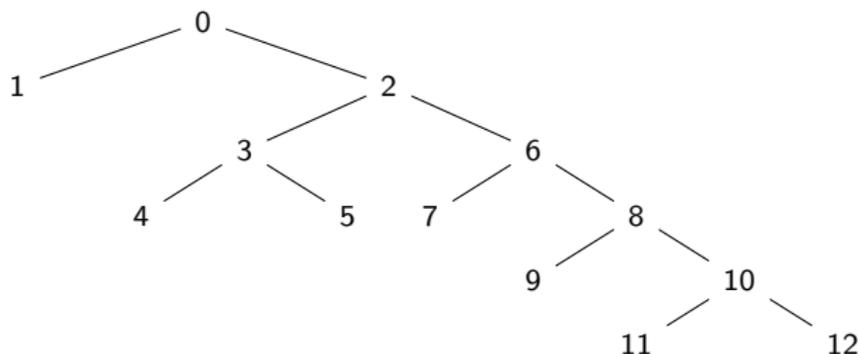
Implémentations simples d'un arbre

Idée

Codage d'une structure complexe par des structures simples (listes, tableaux...) (peut-être pas optimal mais possible)

- 1 une liste T : le père du sommet k est le sommet $T[k]$.

Exemple : $T = [/, 0, 0, 2, 3, 3, 2, 6, 6, 8, 8, 10, 10]$

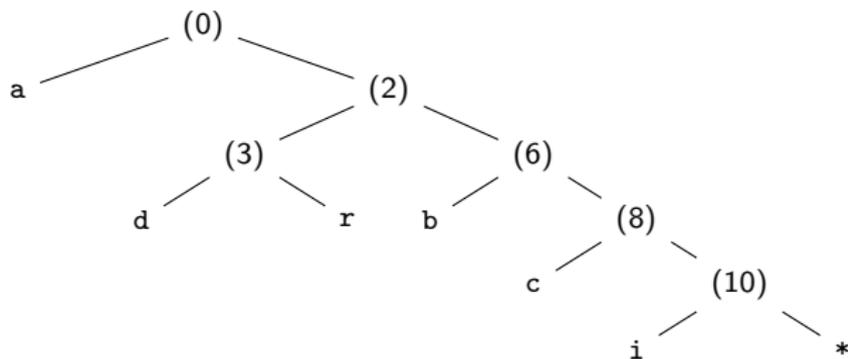


Codages simples d'un arbre dans un ordinateur

② (arbres binaires) tableau à quatre lignes

Exemple :

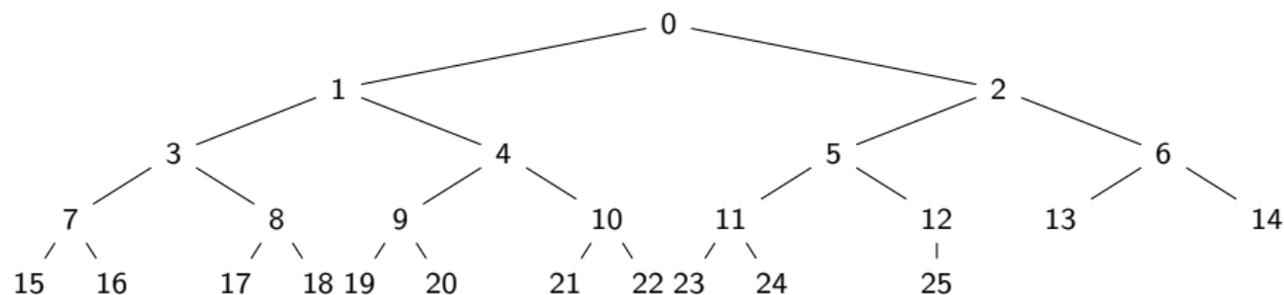
Sommet	0	1	2	3	4	5	6	7	8	9	10	11	12
Fils G	1	/	3	4	/	/	7	/	9	/	11	/	/
Fils D	2	/	6	5	/	/	8	/	10	/	12	/	/
Label	""	a	""	""	d	r	""	b	""	c	""	i	*



Codages simples d'un arbre dans un ordinateur

- ③ Une liste peut être considérée comme un arbre binaire presque complet (*lecture* de haut en bas et de gauche à droite).

Exemple : $[0,1,2,3,\dots,24,25]$



Usage : tri par tas (*heapsort*)