

# GRAPHES ET LANGAGES

## CHAPITRE 4

### ARBRES

Leo Donati    Noëlle Stolfi

Université de Nice Sophia Antipolis  
IUT Nice Côte d'Azur  
DUT Informatique

2015-2016



# CHAPITRE 4 : ARBRES

## 1 ARBRES

- Définition
- Vocabulaire
- Arbre de recouvrement

## 2 ARBRES ENRACINÉS

- Définition
- Vocabulaire
- Taille

## 3 ARBRES BINAIRES

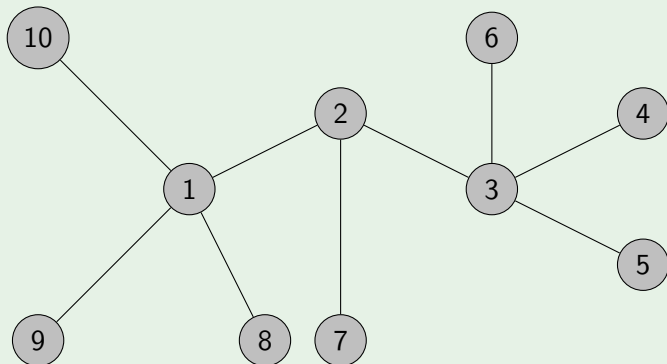
- Définition
- Notation binaire des sommets
- Codage de Huffman

# ARBRES

## DÉFINITION

Un arbre est un graphe non orienté connexe et sans cycle.

## EXEMPLE D'ARBRE



# PROPRIÉTÉS

SI  $G = (V, E, \gamma)$  EST UN ARBRE ALORS

- $G$  est simple
- le degré de  $G$  est égal à l'ordre de  $G$  moins un.
- chaque arête est un **pont**, c'est à dire qu'en le supprimant le graphe n'est plus connexe.
- étant donnés deux sommets  $s$  et  $t$ , il existe une et une seule chaîne allant de  $s$  à  $t$ .

## THÉORÈME

SI  $G$  est un graphe non orienté, simple et connexe, d'ordre  $n$  avec  $n - 1$  arêtes, alors  $G$  est un arbre.

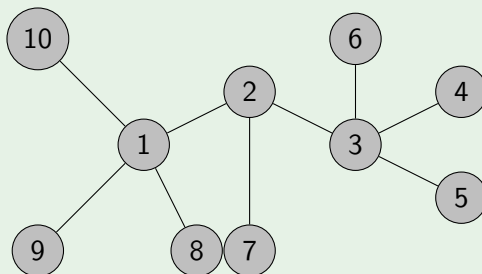
# VOCABULAIRE

SOIT  $G$  UN ARBRE, ALORS

- les arêtes s'appellent des **branches**.
- les sommets de degré 1 s'appellent des **feuilles**.
- les autres sommets s'appellent des **nœuds**.

# EXEMPLE

## SUR L'EXEMPLE



- Les feuilles sont 4, 5, 6, 7, 8, 9, 10
- Les nœuds sont 1, 2, 3
- Le diamètre de l'arbre est 4

# ARBRE COUVRANT

## DÉFINITION

Soit  $G = (V, E, \gamma)$  un graphe simple non orienté. Un **arbre couvrant** de  $G$  est un sous-graphe de  $G$  qui est un arbre contenant chaque sommet de  $G$ .

## REMARQUE

- Il faut évidemment que  $G$  soit connexe pour que l'arbre de recouvrement existe.  
Mais si c'est le cas son existence est assurée mais pas son **unicité**.
- Si le graphe est hamiltonien, alors le cycle hamiltonien (sans la dernière arête fermante) est un exemple d'arbre de recouvrement.

# ALGORITHME DE CONSTRUCTION

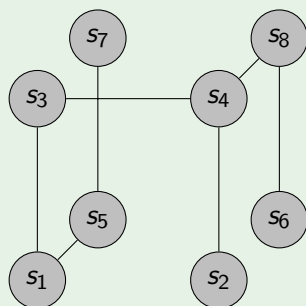
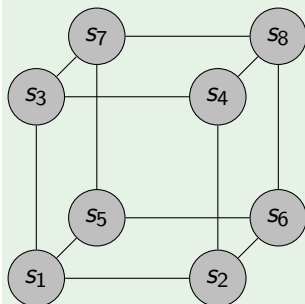
## CONSTRUCTION D'UN ARBRE COUVRANT

- On choisit arbitrairement un sommet  $s$  de  $G$
- à partir de  $s$  on construit une chaîne simple en ajoutant des arêtes de  $G$
- si la chaîne ainsi construite contient toutes les arêtes de  $G$  alors on a un arbre de recouvrement.
- sinon on revient à l'avant dernier sommet et on part de ce sommet pour construire une nouvelle chaîne simple sans repasser par aucun sommet déjà utilisé, si ce n'est pas possible on revient au sommet précédent...
- chaque fois qu'on se bloque on réitère ce processus.



## EXEMPLE

## ARBRE COUVRANT POUR LE GRAPHE DU CUBE

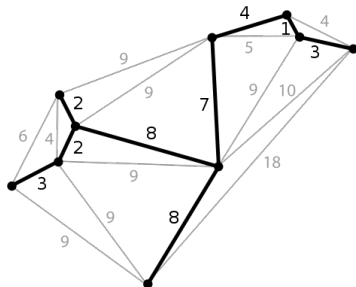


Le diamètre est passé de 3 à 6. Est-ce qu'on peut avoir le même diamètre ?

# ARBRE COUVRANT DE POIDS MINIMAL

## DÉFINITION

Si  $G = (E, V, \gamma, \omega)$  est un graphe non orienté simple et **valué**, un **arbre couvrant de poids minimal** (*minimum spanning tree*) est un arbre couvrant de  $G$  dont la somme des poids des arêtes est minimal.



# ALGORITHME DE PRIM

## PRINCIPE DE L'ALGORITHME (1957)

- Initialisation : avec un sommet quelconque
- Récursivement : si on a déjà un arbre minimal avec  $n$  sommets, on fait la liste de toutes les arêtes reliant un sommet non sélectionné à un sommet sélectionné et on choisit l'arête de poids minimal que l'on ajoute à l'arbre minimal.
- L'algorithme termine quand tous les sommets sont sélectionnés.

# CHAPITRE 4 : ARBRES

## 1 ARBRES

- Définition
- Vocabulaire
- Arbre de recouvrement

## 2 ARBRES ENRACINÉS

- Définition
- Vocabulaire
- Taille

## 3 ARBRES BINAIRES

- Définition
- Notation binaire des sommets
- Codage de Huffman

# ARBRES ENRACINÉS

## DÉFINITION

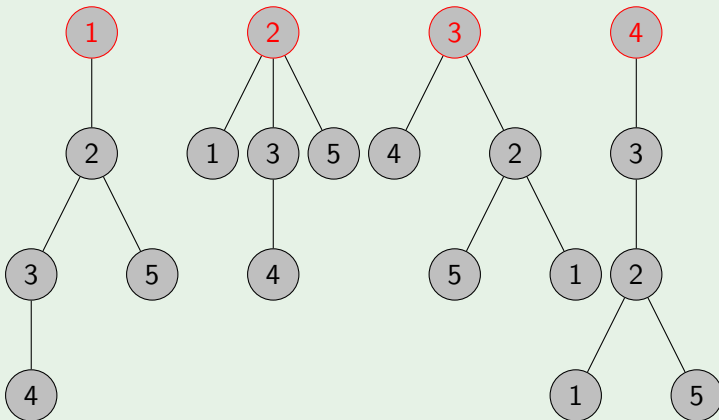
Un arbre **enraciné** est un arbre dans lequel on sélectionne un sommet que l'on nomme **racine** de l'arbre.

## REMARQUE

Le même arbre peut être enraciné de plusieurs façons non isomorphes.

# EXEMPLES

## LE MÊME ARBRE, PLUSIEURS RACINES



# NIVEAU ET HAUTEUR

## DÉFINITIONS

Un arbre enraciné possède une orientation naturelle par rapport à sa racine  $R$  :

- le **niveau** d'un sommet  $s$  est la distance entre  $s$  et la racine  $R$  ;
- la **hauteur** d'un arbre enraciné est le niveau le plus élevé atteint par un sommet. Notation :  $h(G)$

## REMARQUE

- La racine est le seul sommet de niveau 0
- La hauteur d'un arbre enraciné est toujours inférieure ou égale à son diamètre.

# VOCABULAIRE

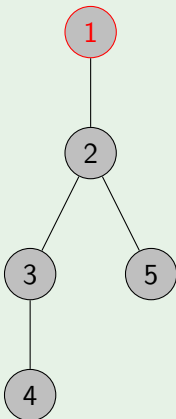
SI  $(G, R)$  EST UN ARBRE ENRACINÉ, ALORS

- Soient  $s$  et  $t$  deux sommets adjacents ; le **père** est le sommet plus proche de la racine, et le **fils** est celui qui est le plus éloigné.
- Les **ancêtres** d'un sommet  $s$  sont tous les nœuds par lesquels on passe en suivant le chemin allant de  $s$  vers la racine.
- Les **descendants** d'un sommet  $t$  sont tous les nœuds dont  $t$  est un ancêtre



# EXEMPLE

## EXEMPLE



- hauteur de l'arbre = 3
- 1 est le père de 2
- 3 et 5 sont les fils de 2
- les ascendants de 4 sont 1, 2, 3
- les descendants de 2 sont 3, 4 et 5

# TAILLE D'UN ARBRE

## PROPOSITION

- Si le sommet  $s$  est de niveau  $d$  alors il a  $d$  ancêtres
- Si dans l'arbre enraciné,  $n$  est le nombre maximum de fils que peut avoir un nœud et  $h$  est la hauteur de l'arbre, alors l'ordre du graphe est au maximum

$$o(G) \leq 1 + n + n^2 + n^3 + \dots + n^h < n^{h+1}$$

# CHAPITRE 4 : ARBRES

## 1 ARBRES

- Définition
- Vocabulaire
- Arbre de recouvrement

## 2 ARBRES ENRACINÉS

- Définition
- Vocabulaire
- Taille

## 3 ARBRES BINAIRES

- Définition
- Notation binaire des sommets
- Codage de Huffman

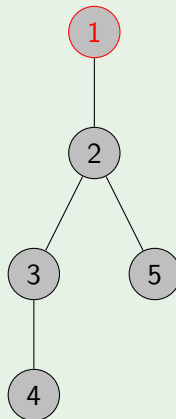
# ARBRES BINAIRES

## DÉFINITION

Un **arbre binaire** est un arbre enraciné dans lequel chaque nœud a au maximum deux fils. De plus ces deux fils sont ordonnés :

- à gauche on place le premier fils : le fils aîné
- à droite on place le second fils : le fils cadet

## EXEMPLE

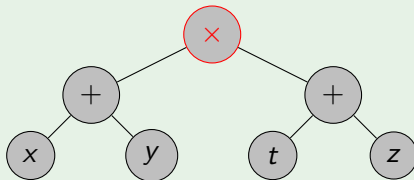


# UTILISATION

## EXEMPLE D'APPLICATIONS DES ARBRES BINAIRES

- arbre de recherche
- arbre d'une formule

FORMULE  $(x + y)(z + t)$



# PROPRIÉTÉS

## PROPRIÉTÉS DES ARBRES BINAIRES

- Le degré de chaque nœud est au maximum 3 (un père et deux fils)
- Si  $h$  est la hauteur de l'arbre, le nombre de sommets est au maximum  $2^{h+1}$

# NOTATION BINAIRE DES SOMMETS

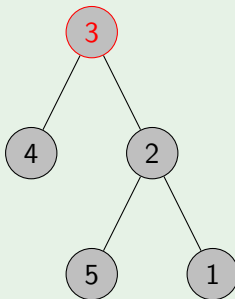
## PRINCIPE

Dans un arbre binaire on peut attacher à chaque noeud ou feuille une étiquette binaire qui décrit précisément la position du sommet dans l'arborescence :

- la racine est dénotée par le mot vide
- les nœuds de niveau  $d$  ont une étiquette binaire avec  $d$  bits
- l'étiquette binaire d'un fils s'obtient à partir de l'étiquette du père en ajoutant à droite soit un 0 pour le fils aîné soit un 1 pour le fils cadet

## EXEMPLE DE NOTATION BINAIRE

### EXEMPLE

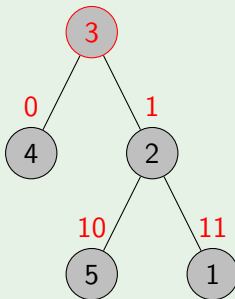


Chaque nœud ou feuille est identifié de façon unique par son étiquette binaire qui est aussi un **codage** du chemin à suivre pour y arriver à partir de la racine.



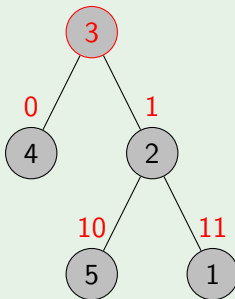
# EXEMPLE DE NOTATION BINAIRE

## EXEMPLE



# EXEMPLE DE NOTATION BINAIRE

## EXEMPLE

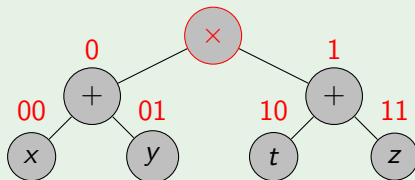


Chaque nœud ou feuille est identifié de façon unique par son étiquette binaire qui est aussi un **codage** du chemin à suivre pour y arriver à partir de la racine.

# ARBRE BINAIRE D'UNE FORMULE

## EXEMPLE

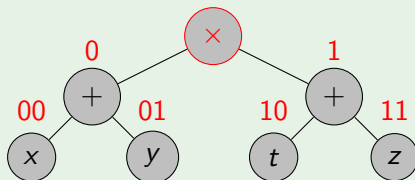
Si l'on a une formule mathématique, on peut ainsi coder les opérateurs et les opérandes par leur étiquette binaire :



# NOTATION POLONAISE

## DÉFINITION

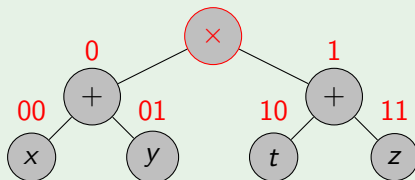
La **notation polonaise** d'une formule mathématique, consiste à écrire ses éléments dans l'ordre lexicographique de l'étiquette binaire des sommets.



# NOTATION POLONAISE

## DÉFINITION

La **notation polonaise** d'une formule mathématique, consiste à écrire ses éléments dans l'ordre lexicographique de l'étiquette binaire des sommets.



En écriture polonaise donne :  $x + x y + t z$   
ce qui correspond à une écriture **préfixée** car on met l'opérateur avant les opérandes, et cela récursivement.

# TRAVERSÉE D'UN ARBRE BINAIRE

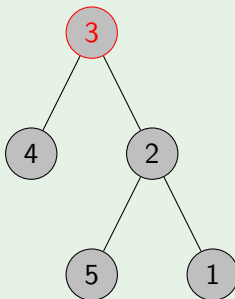
## COMMENT S'Y DÉPLACER ?

Il y a essentiellement 3 façons classiques de traverser un arbre binaire, c'est à dire de passer par tous les sommets :

- parcours **préfixe** : c'est celui donné par l'ordre lexicographique sur les étiquettes binaires ; d'abord la racine puis récursivement le sous arbre gauche puis récursivement le sous-arbre droit
- parcours **postfixe** : d'abord récursivement le sous-arbre gauche, puis le sous-arbre droit et enfin la racine
- parcours **infixe** : d'abord récursivement le sous-arbre gauche, puis la racine, puis récursivement le sous-arbre droit.

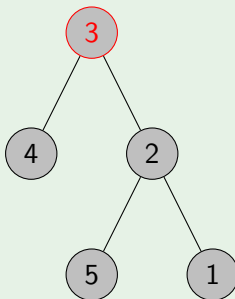
# EXEMPLES DE TRAVERSÉE

## EXEMPLE



# EXEMPLES DE TRAVERSÉE

## EXEMPLE

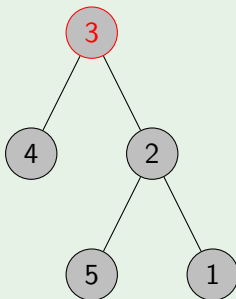


- Parcours préfixe : 3 - 4 - 2 - 5 - 1



# EXEMPLES DE TRAVERSÉE

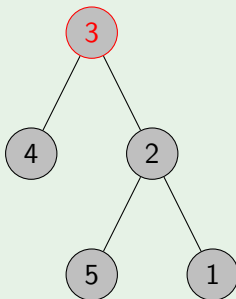
## EXEMPLE



- Parcours préfixe : 3 - 4 - 2 - 5 - 1
- Parcours postfixe : 4 - 5 - 1 - 2 - 3

# EXEMPLES DE TRAVERSÉE

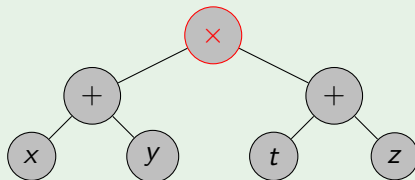
## EXEMPLE



- Parcours préfixe : 3 - 4 - 2 - 5 - 1
- Parcours postfixe : 4 - 5 - 1 - 2 - 3
- Parcours infixe : 4 - 3 - 5 - 2 - 1

# SUR UNE FORMULE MATHÉMATIQUE

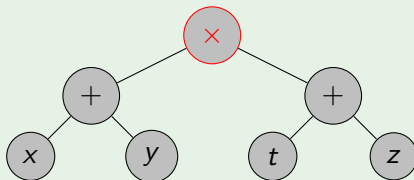
## EXEMPLE



- Parcours préfixe :  $x + y + t z$  : écriture polonaise

# SUR UNE FORMULE MATHÉMATIQUE

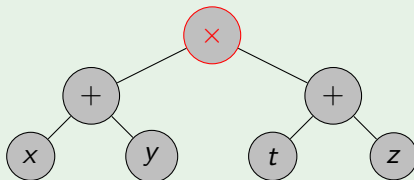
## EXEMPLE



- Parcours préfixe :  $x + y + t z$  : écriture polonaise
- Parcours postfixe :  $x y + t z + x$  : **écriture polonaise inversée** (appelée aussi RPN pour Reverse Polish Notation)

# SUR UNE FORMULE MATHÉMATIQUE

## EXEMPLE



- Parcours préfixe :  $\times + x y + t z$  : écriture polonaise
- Parcours postfixe :  $x y + t z + \times$  : **écriture polonaise inversée** (appelée aussi RPN pour Reverse Polish Notation)
- Parcours infixe :  $x + y \times t + z$  : écriture classique des formules qui nécessite obligatoirement d'utiliser des parenthèses :  $(x + y) \times (t + z)$

# CODAGE DE HUFFMAN

## DAVID HUFFMAN (1952)

Le codage de Huffman est un algorithme de **compression sans perte** qui utilise un codage à longueur variable d'un symbole selon sa fréquence dans le texte à compresser.

## PRINCIPE

- On parcourt le texte pour connaître le nombre d'occurrence de chaque symbole
- On construit **l'arbre des fréquences** qui est binaire
- La notation binaire des feuilles est utilisée pour coder le texte

# ARBRE DES FRÉQUENCES

## ARBRE BINAIRE

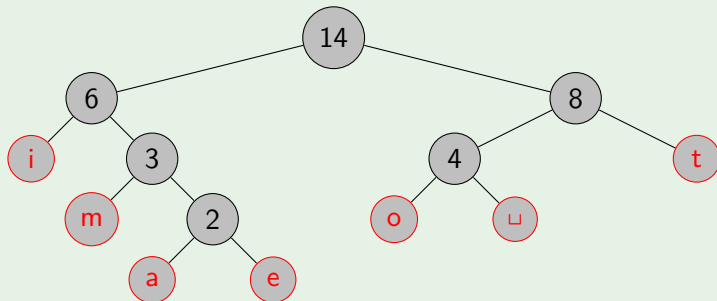
- les feuilles sont les symboles dont le poids est donné par leur nombre d'occurrence
- l'arbre binaire est construit en prenant les deux arbres binaires de poids le plus faible et en construisant un nouvel arbre en donnant au père la somme des poids des fils
- on place à gauche le fils qui a le poids le plus faible (ou à égalité celui qui arrive avant dans l'ordre lexicographique)

# EXEMPLE D'ARBRE DES FRÉQUENCES

TEXTE = 'TOTO AIME TITI'

Nombre d'occurrences :

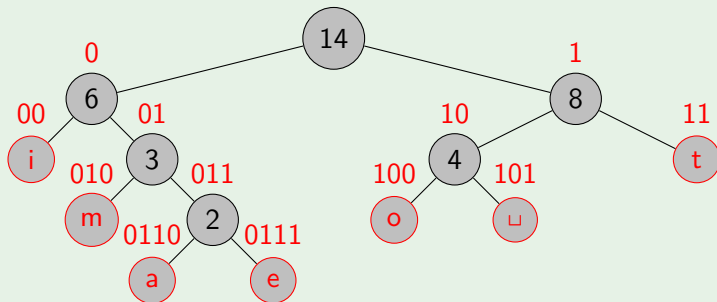
a	e	m	o	□	i	t
1	1	1	2	2	3	4





# EXEMPLE DE CODAGE

## CODAGE



"toto\_aime\_titi" → 1110011100101011000010011110111001100