

Intelligence Artificielle 2

Cours 1 - Arbres

Steve Lévesque, Tous droits réservés © où applicables

Table des matières

1 Arbre

- Définition
- Arbre graphique avec Graphviz

2 Techniques de traverse d'arbres

- Explication de la vision d'un noeud par rapport à la traverse
- Preorder (Avant)
- Inorder (Position actuelle)
- Postorder (Après)

Définition

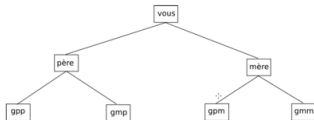
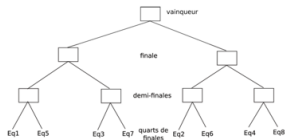
L'arbre est une structure de données hiérarchique (non-linéaire de manière conceptuelle). Ceci-dit, la structure peut être stockée dans un tableau (linéaire).

Un arbre binaire est un arbre avec un maximum de 2 branches à chaque nœud.

Les composantes principales d'un arbre :

- Racine
- Notion de relation parent/enfant
- Stockage possiblement arbitraire (linéaire ou non) :
 - Dans une liste imbriquée avec d'autres listes pour les enfants
 - Une simple liste, dépilée par une formule qui permet de reconstruire l'arbre d'origine (doit être un arbre binaire complet et équilibré)

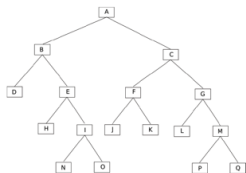
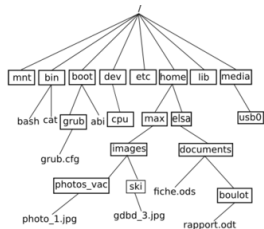
Définition - Graphique structure régulière



Arbre 1



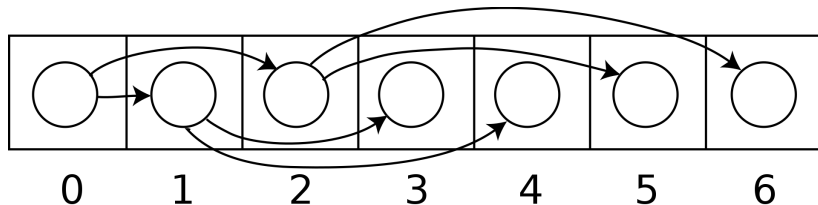
Arbre 2



Définition - Stockage dans une liste

Calcul : Pour un parent i , enfant gauche : $2i + 1$, enfant droite : $2i + 2$.

Exemple : $i = 0$, enfant gauche = $2 * 0 + 1 = 1$, enfant droite = $2 * 0 + 2 = 2$.



Définition - Vocabulaire additionnel

- Racine, Noeud, Parent (Noeud supérieur), Enfant, Feuille (enfants finaux les plus bas)
- 2 enfants est un lien de parenté (“siblings”)
- Sous-arbre (petit arbre en dessous d'un noeud racine)
- NIL (Sous-arbre vide, nil veut dire nul)
- Arête (représentation graphique)
- Nombre de niveaux :
 - Profondeur d'un noeud/feuille
 - Hauteur d'un arbre

Définition - Vocabulaire additionnel

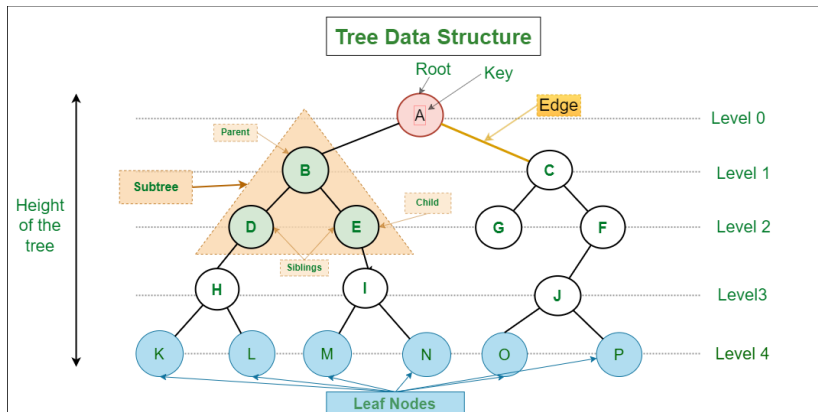


Figure: <https://www.geeksforgeeks.org/tree-traversal-techniques-in-python/>

Arbre graphique avec Graphviz

Exécuter et générer le PDF à partir du code de la diapo suivante.

Il faut installer Grapgviz sur Windows/Linux/Mac et ensuite le module graphviz de Python.

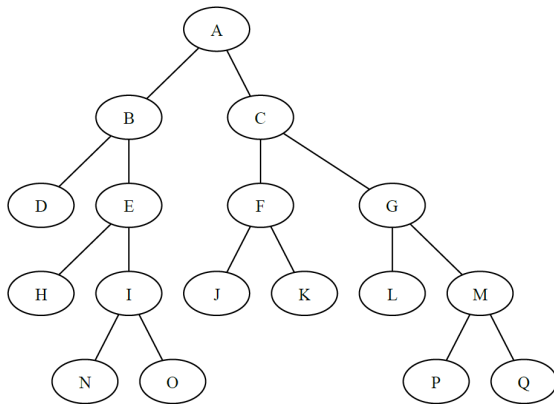
Essayez de le faire vous même sans démo de ma part, les installations sont expliquées sur Internet et le principe change beaucoup au fil du temps.

Arbre graphique avec Graphviz

```
1 import graphviz
2
3 dot = graphviz.Graph(comment='Graphviz ex1')
4
5 dot.edges(['AB', 'AC', 'BD', 'BE', 'CF', 'CG', 'EH', 'EI',
6           'FJ', 'FK', 'GL', 'GM', 'IN', 'IO', 'MP', 'MQ'])
7
8 dot.render('graphviz-output/1-graphviz_pdf.gv')
```

Arbre graphique avec Graphviz

PDF généré à partir du code précédent :



Techniques de traverse d'arbres

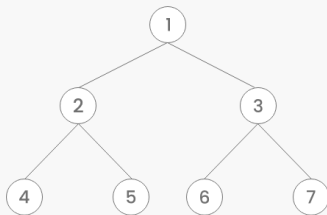
Les arbres peuvent être traversés de 3 différentes manières standards.

- Preorder (avant)
- Inorder (à la position ordonnée/actuelle)
- Postorder (après)

On traverse **toujours** de gauche à droite. Ceci est une technique de traverse “depth first” (profondeur en premier).

Techniques de traverse d'arbres

Tree Traversal Techniques



Inorder Traversal

4	2	5	1	6	3	7
---	---	---	---	---	---	---

Preorder Traversal

1	2	4	5	3	6	7
---	---	---	---	---	---	---

Postorder Traversal

4	5	2	6	7	3	1
---	---	---	---	---	---	---

Figure: <https://www.geeksforgeeks.org/tree-traversal-techniques-in-python/>

Techniques de traverse d'arbres - Explication de la vision d'un noeud par rapport à la traverse

Il faut le voir comme étant la règle dont on considère lorsqu'on le touche (traverse) le noeud lors de notre traversé d'arbre.

- En rouge, lorsqu'on le traverse en le touchant à gauche (avant) : Preorder
- En bleu, lorsqu'on le traverse en le touchant en dessous (position actuelle) : Inorder
- En jaune, lorsqu'on le traverse en le touchant à droite (après) : Postorder

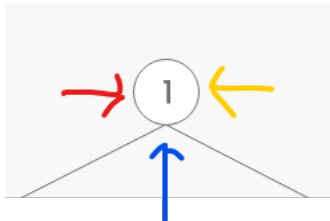


Figure: <https://www.geeksforgeeks.org/tree-traversal-techniques-in-python/>

Techniques de traverse d'arbres - Preorder (Avant)

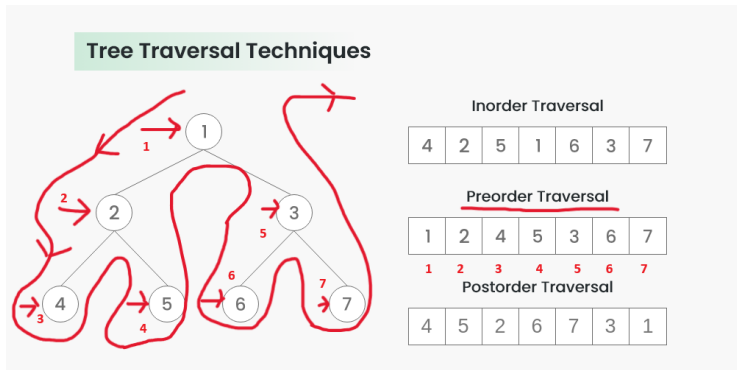


Figure: <https://www.geeksforgeeks.org/tree-traversal-techniques-in-python/>

Techniques de traverse d'arbres - Inorder (Position actuelle)

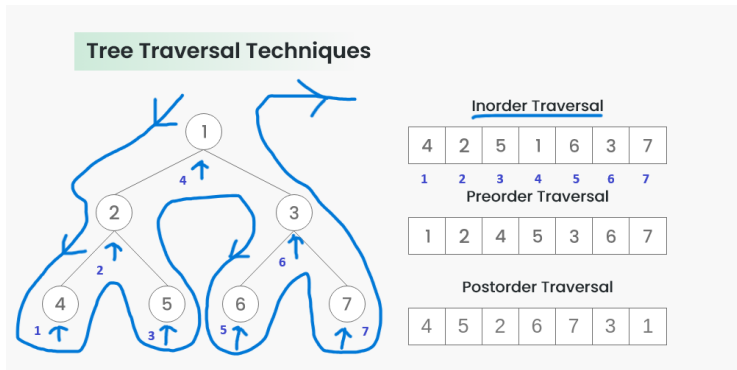


Figure: <https://www.geeksforgeeks.org/tree-traversal-techniques-in-python/>

Techniques de traverse d'arbres - Postorder (Après)

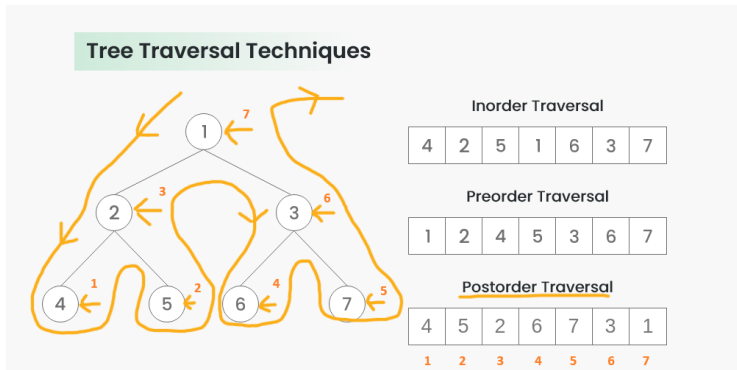


Figure: <https://www.geeksforgeeks.org/tree-traversal-techniques-in-python/>

Bibliographie

- `https://www.geeksforgeeks.org/
tree-traversal-techniques-in-python/`