

Programmation Fonctionnelle

Séance 6 : Parcours d'arbres, arbres binaires de recherche.

Alexandros Singh

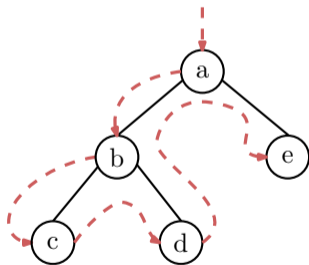
Université Paris 8

18 novembre 2023

Un parcours d'arbre est un algorithme permettant de visiter tous les sommets d'un graphe. Nous étudierons deux des principaux types d'algorithmes de ce type :

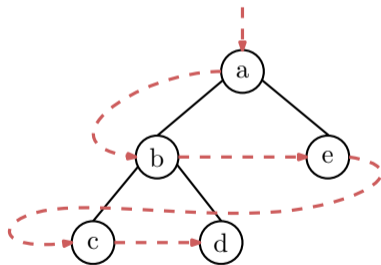
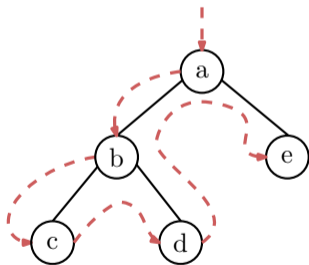
Un parcours d'arbre est un algorithme permettant de visiter tous les sommets d'un graphe. Nous étudierons deux des principaux types d'algorithmes de ce type :

- Parcours en profondeur (DFS)



Un parcours d'arbre est un algorithme permettant de visiter tous les sommets d'un graphe. Nous étudierons deux des principaux types d'algorithmes de ce type :

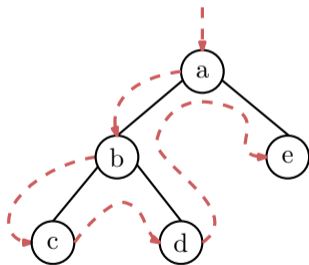
- Parcours en profondeur (DFS)
- Parcours en largeur (BFS)



Un parcours d'arbre est un algorithme permettant de visiter tous les sommets d'un graphe. Nous étudierons deux des principaux types d'algorithmes de ce type :

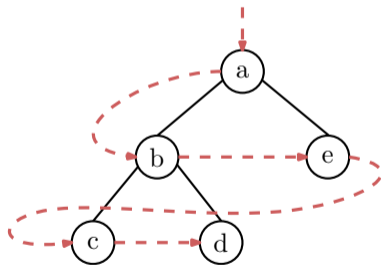
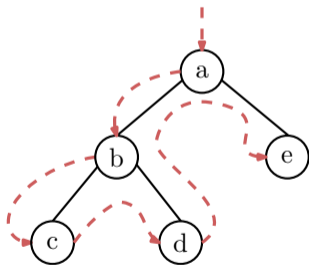
Un parcours d'arbre est un algorithme permettant de visiter tous les sommets d'un graphe. Nous étudierons deux des principaux types d'algorithmes de ce type :

- Parcours en profondeur (DFS)



Un parcours d'arbre est un algorithme permettant de visiter tous les sommets d'un graphe. Nous étudierons deux des principaux types d'algorithmes de ce type :

- Parcours en profondeur (DFS)
- Parcours en largeur (BFS)



Le parcours en profondeur est naturellement décrit de manière récursive :

- Si l'arbre d'entrée est vide, une action appropriée est effectuée.

Le parcours en profondeur est naturellement décrit de manière récursive :

- Si l'arbre d'entrée est vide, une action appropriée est effectuée.
- Sinon, les trois actions suivantes sont effectuées dans un certain ordre :

Le parcours en profondeur est naturellement décrit de manière récursive :

- Si l'arbre d'entrée est vide, une action appropriée est effectuée.
- Sinon, les trois actions suivantes sont effectuées dans un certain ordre :
 - A_1 La racine est visitée.

Le parcours en profondeur est naturellement décrit de manière récursive :

- Si l'arbre d'entrée est vide, une action appropriée est effectuée.
- Sinon, les trois actions suivantes sont effectuées dans un certain ordre :
 - A_1 La racine est visitée.
 - A_2 L'algorithme est exécuté récursivement sur le sous-arbre gauche.

Le parcours en profondeur est naturellement décrit de manière récursive :

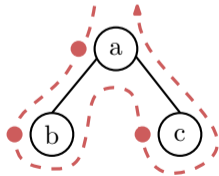
- Si l'arbre d'entrée est vide, une action appropriée est effectuée.
- Sinon, les trois actions suivantes sont effectuées dans un certain ordre :
 - A_1 La racine est visitée.
 - A_2 L'algorithme est exécuté récursivement sur le sous-arbre gauche.
 - A_3 L'algorithme est exécuté récursivement sur le sous-arbre de droite.

Le parcours en profondeur est naturellement décrit de manière récursive :

- Si l'arbre d'entrée est vide, une action appropriée est effectuée.
- Sinon, les trois actions suivantes sont effectuées dans un certain ordre :
 - A_1 La racine est visitée.
 - A_2 L'algorithme est exécuté récursivement sur le sous-arbre gauche.
 - A_3 L'algorithme est exécuté récursivement sur le sous-arbre de droite.

L'ordre dans lequel les actions A_1, A_2, A_3 sont effectuées détermine la variation de l'algorithme :

- A_1, A_2, A_3 : parcours préfixe.

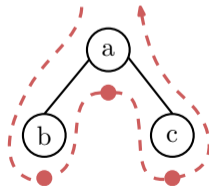


Le parcours en profondeur est naturellement décrit de manière récursive :

- Si l'arbre d'entrée est vide, une action appropriée est effectuée.
- Sinon, les trois actions suivantes sont effectuées dans un certain ordre :
 - A_1 La racine est visitée.
 - A_2 L'algorithme est exécuté récursivement sur le sous-arbre gauche.
 - A_3 L'algorithme est exécuté récursivement sur le sous-arbre de droite.

L'ordre dans lequel les actions A_1, A_2, A_3 sont effectuées détermine la variation de l'algorithme :

- A_1, A_2, A_3 : parcours préfixe.
- A_2, A_1, A_3 : parcours infixé.

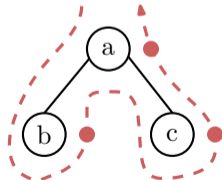


Le parcours en profondeur est naturellement décrit de manière récursive :

- Si l'arbre d'entrée est vide, une action appropriée est effectuée.
- Sinon, les trois actions suivantes sont effectuées dans un certain ordre :
 - A_1 La racine est visitée.
 - A_2 L'algorithme est exécuté récursivement sur le sous-arbre gauche.
 - A_3 L'algorithme est exécuté récursivement sur le sous-arbre de droite.

L'ordre dans lequel les actions A_1, A_2, A_3 sont effectuées détermine la variation de l'algorithme :

- A_1, A_2, A_3 : parcours préfixe.
- A_2, A_1, A_3 : parcours infixé.
- A_2, A_3, A_1 : parcours postfixé.



Le parcours en profondeur est naturellement décrit de manière récursive :

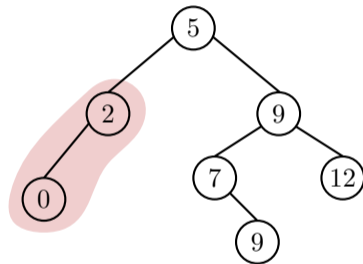
- Si l'arbre d'entrée est vide, une action appropriée est effectuée.
- Sinon, les trois actions suivantes sont effectuées dans un certain ordre :
 - A_1 La racine est visitée.
 - A_2 L'algorithme est exécuté récursivement sur le sous-arbre gauche.
 - A_3 L'algorithme est exécuté récursivement sur le sous-arbre de droite.

L'ordre dans lequel les actions A_1, A_2, A_3 sont effectuées détermine la variation de l'algorithme :

- A_1, A_2, A_3 : parcours préfixe.
- A_2, A_1, A_3 : parcours infixé.
- A_2, A_3, A_1 : parcours postfixé.
- ...

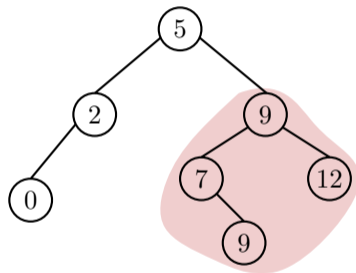
Un arbre de recherche binaire (ABR) est un arbre (unaire-)binaire tel que :

- L'étiquette de la racine est supérieure à toutes celles des sommets du sous-arbre gauche.



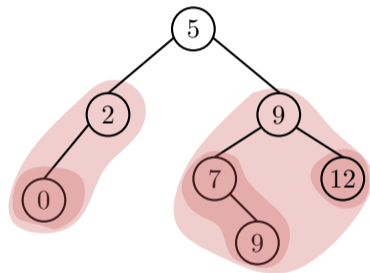
Un arbre de recherche binaire (ABR) est un arbre (unaire-)binaire tel que :

- L'étiquette de la racine est supérieure à toutes celles des sommets du sous-arbre gauche.
- L'étiquette de la racine est supérieure à toutes celles des sommets du sous-arbre droit.



Un arbre de recherche binaire (ABR) est un arbre (unaire-)binaire tel que :

- L'étiquette de la racine est supérieure à toutes celles des sommets du sous-arbre gauche.
- L'étiquette de la racine est supérieure à toutes celles des sommets du sous-arbre droit.
- Les sous-arbres sont eux-mêmes des ABR.



L'insertion d'une valeur dans un arbre de recherche binaire s'effectue comme suit :

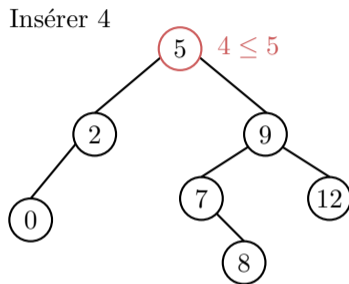
- Si l'arbre est vide, un nouveau sommet est créé avec la valeur souhaitée comme étiquette.

L'insertion d'une valeur dans un arbre de recherche binaire s'effectue comme suit :

- Si l'arbre est vide, un nouveau sommet est créé avec la valeur souhaitée comme étiquette.
- Sinon, la valeur est comparée à celle de la racine :

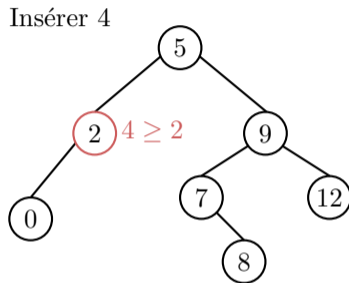
L'insertion d'une valeur dans un arbre de recherche binaire s'effectue comme suit :

- Si l'arbre est vide, un nouveau sommet est créé avec la valeur souhaitée comme étiquette.
- Sinon, la valeur est comparée à celle de la racine :
 - Si elle est plus petite, l'algorithme est exécuté de manière réursive sur le sous-arbre gauche.



L'insertion d'une valeur dans un arbre de recherche binaire s'effectue comme suit :

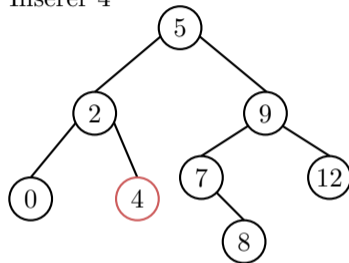
- Si l'arbre est vide, un nouveau sommet est créé avec la valeur souhaitée comme étiquette.
- Sinon, la valeur est comparée à celle de la racine :
 - Si elle est plus petite, l'algorithme est exécuté de manière récursive sur le sous-arbre gauche.
 - Si elle est plus grande, l'algorithme est exécuté de manière récursive sur le sous-arbre droit.



L'insertion d'une valeur dans un arbre de recherche binaire s'effectue comme suit :

- Si l'arbre est vide, un nouveau sommet est créé avec la valeur souhaitée comme étiquette.
- Sinon, la valeur est comparée à celle de la racine :
 - Si elle est plus petite, l'algorithme est exécuté de manière réursive sur le sous-arbre gauche.
 - Si elle est plus grande, l'algorithme est exécuté de manière réursive sur le sous-arbre droit.

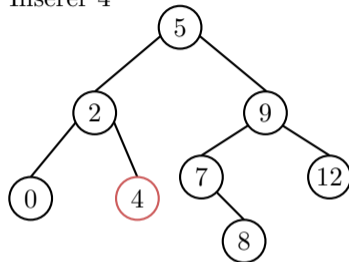
Insérer 4



L'insertion d'une valeur dans un arbre de recherche binaire s'effectue comme suit :

- Si l'arbre est vide, un nouveau sommet est créé avec la valeur souhaitée comme étiquette.
- Sinon, la valeur est comparée à celle de la racine :
 - Si elle est plus petite, l'algorithme est exécuté de manière récursive sur le sous-arbre gauche.
 - Si elle est plus grande, l'algorithme est exécuté de manière récursive sur le sous-arbre droit.
 - S'il est égal, nous pouvons choisir soit de ne pas l'inclure, soit d'effectuer un appel récursif.^a

Insérer 4



a. la variation précise choisie dépend de l'utilisation que nous souhaitons faire des ABR

L'insertion d'une valeur dans un arbre binaire de recherche s'effectue comme suit :

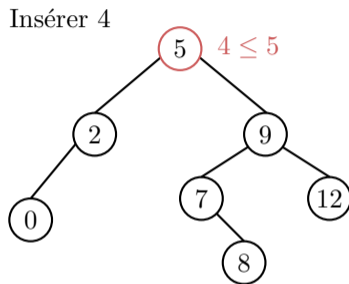
- Si l'arbre est vide, un nouveau sommet est créé avec la valeur souhaitée comme étiquette.

L'insertion d'une valeur dans un arbre binaire de recherche s'effectue comme suit :

- Si l'arbre est vide, un nouveau sommet est créé avec la valeur souhaitée comme étiquette.
- Sinon, la valeur est comparée à celle de la racine :

L'insertion d'une valeur dans un arbre binaire de recherche s'effectue comme suit :

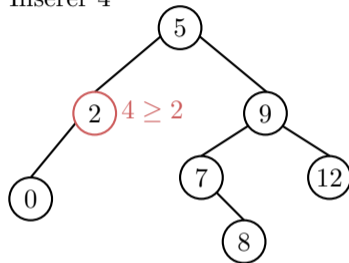
- Si l'arbre est vide, un nouveau sommet est créé avec la valeur souhaitée comme étiquette.
- Sinon, la valeur est comparée à celle de la racine :
 - Si elle est plus petite, l'algorithme est exécuté de manière récursive sur le sous-arbre gauche.



L'insertion d'une valeur dans un arbre binaire de recherche s'effectue comme suit :

- Si l'arbre est vide, un nouveau sommet est créé avec la valeur souhaitée comme étiquette.
- Sinon, la valeur est comparée à celle de la racine :
 - Si elle est plus petite, l'algorithme est exécuté de manière récursive sur le sous-arbre gauche.
 - Si elle est plus grande, l'algorithme est exécuté de manière récursive sur le sous-arbre droit.

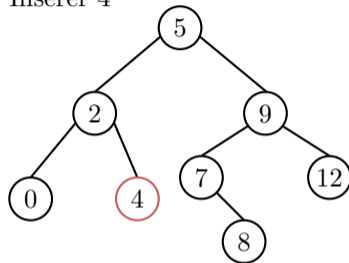
Insérer 4



L'insertion d'une valeur dans un arbre binaire de recherche s'effectue comme suit :

- Si l'arbre est vide, un nouveau sommet est créé avec la valeur souhaitée comme étiquette.
- Sinon, la valeur est comparée à celle de la racine :
 - Si elle est plus petite, l'algorithme est exécuté de manière récursive sur le sous-arbre gauche.
 - Si elle est plus grande, l'algorithme est exécuté de manière récursive sur le sous-arbre droit.

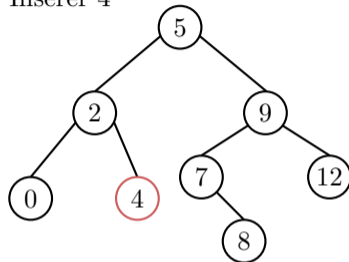
Insérer 4



L'insertion d'une valeur dans un arbre binaire de recherche s'effectue comme suit :

- Si l'arbre est vide, un nouveau sommet est créé avec la valeur souhaitée comme étiquette.
- Sinon, la valeur est comparée à celle de la racine :
 - Si elle est plus petite, l'algorithme est exécuté de manière récursive sur le sous-arbre gauche.
 - Si elle est plus grande, l'algorithme est exécuté de manière récursive sur le sous-arbre droit.
 - S'il est égal, nous pouvons choisir soit de ne pas l'inclure, soit d'effectuer un appel récursif. ^a

Insérer 4



a. la variation précise choisie dépend de l'utilisation que nous souhaitons faire des ABR

La recherche d'une valeur dans un arbre binaire de recherche s'effectue de la manière suivante :

- Si l'arbre est vide, l'algorithme répond négativement.

La suppression d'une valeur d'un arbre binaire de recherche s'effectue de la manière suivante :

La recherche d'une valeur dans un arbre binaire de recherche s'effectue de la manière suivante :

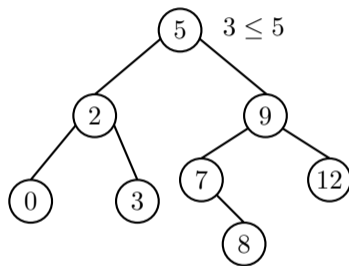
- Si l'arbre est vide, l'algorithme répond négativement.
- Sinon, la valeur est comparée à celle de la racine :

La suppression d'une valeur d'un arbre binaire de recherche s'effectue de la manière suivante :

La recherche d'une valeur dans un arbre binaire de recherche s'effectue de la manière suivante :

- Si l'arbre est vide, l'algorithme répond négativement.
- Sinon, la valeur est comparée à celle de la racine :
 - Si elle est plus petite, l'algorithme est exécuté de manière réursive sur le sous-arbre gauche.

Recherche de 3

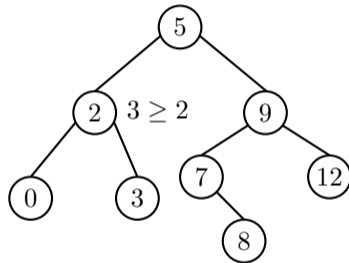


La suppression d'une valeur d'un arbre binaire de recherche s'effectue de la manière suivante :

La recherche d'une valeur dans un arbre binaire de recherche s'effectue de la manière suivante :

- Si l'arbre est vide, l'algorithme répond négativement.
- Sinon, la valeur est comparée à celle de la racine :
 - Si elle est plus petite, l'algorithme est exécuté de manière récursive sur le sous-arbre gauche.
 - Si elle est plus grande, l'algorithme est exécuté de manière récursive sur le sous-arbre droit.

Recherche de 3



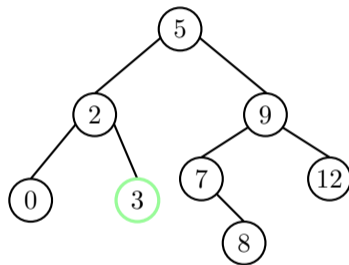
La suppression d'une valeur d'un arbre binaire de recherche s'effectue de la manière suivante :

La recherche d'une valeur dans un arbre binaire de recherche s'effectue de la manière suivante :

- Si l'arbre est vide, l'algorithme répond négativement.
- Sinon, la valeur est comparée à celle de la racine :
 - Si elle est plus petite, l'algorithme est exécuté de manière réursive sur le sous-arbre gauche.
 - Si elle est plus grande, l'algorithme est exécuté de manière réursive sur le sous-arbre droit.
 - S'il est égal, l'algorithme répond positivement, la valeur a été trouvée.

La suppression d'une valeur d'un arbre binaire de recherche s'effectue de la manière suivante :

Recherche de 3



Opérations sur les ABR - Suppression

La suppression d'une valeur d'un arbre binaire de recherche s'effectue de la manière suivante :

- Si l'arbre ne comporte qu'une feuille, nous la supprimons.

Supprimer 2



Opérations sur les ABR - Suppression

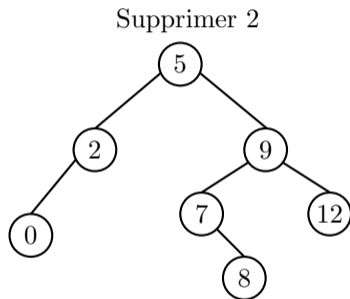
La suppression d'une valeur d'un arbre binaire de recherche s'effectue de la manière suivante :

- Si l'arbre ne comporte qu'une feuille, nous la supprimons.

Supprimer 2

La suppression d'une valeur d'un arbre binaire de recherche s'effectue de la manière suivante :

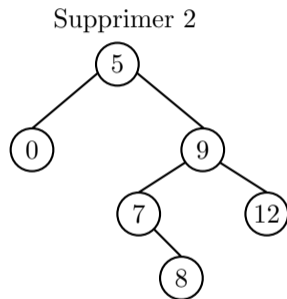
- Si l'arbre ne comporte qu'une feuille, nous la supprimons.
- Si la valeur à supprimer est l'étiquette d'un sommet qui a un enfant unique, nous remplaçons ce sommet par son enfant.



Opérations sur les ABR - Suppression

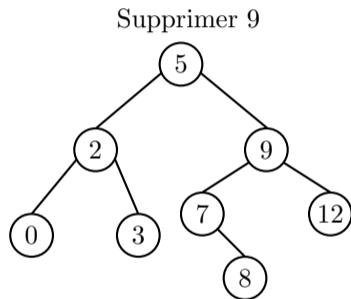
La suppression d'une valeur d'un arbre binaire de recherche s'effectue de la manière suivante :

- Si l'arbre ne comporte qu'une feuille, nous la supprimons.
- Si la valeur à supprimer est l'étiquette d'un sommet qui a un enfant unique, nous remplaçons ce sommet par son enfant.



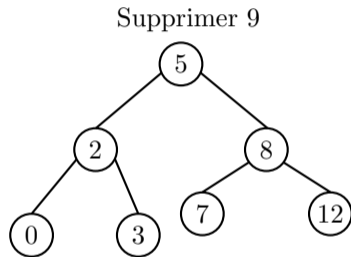
La suppression d'une valeur d'un arbre binaire de recherche s'effectue de la manière suivante :

- Si l'arbre ne comporte qu'une feuille, nous la supprimons.
- Si la valeur à supprimer est l'étiquette d'un sommet qui a un enfant unique, nous remplaçons ce sommet par son enfant.
- Si la valeur que nous voulons supprimer apparaît comme l'étiquette d'un sommet v avec deux enfants, nous remplaçons l'étiquette de v par celle de son prédécesseur p dans la traversée infixe de l'arbre, et nous supprimons p .



La suppression d'une valeur d'un arbre binaire de recherche s'effectue de la manière suivante :

- Si l'arbre ne comporte qu'une feuille, nous la supprimons.
- Si la valeur à supprimer est l'étiquette d'un sommet qui a un enfant unique, nous remplaçons ce sommet par son enfant.
- Si la valeur que nous voulons supprimer apparaît comme l'étiquette d'un sommet v avec deux enfants, nous remplaçons l'étiquette de v par celle de son prédécesseur p dans la traversée infixe de l'arbre, et nous supprimons p .



- Le prédécesseur d'un sommet dans l'ordre infixe est défini de manière équivalente comme étant la feuille avec l'étiquette maximale du sous-arbre de gauche.

- Le prédécesseur d'un sommet dans l'ordre infixe est défini de manière équivalente comme étant la feuille avec l'étiquette maximale du sous-arbre de gauche.
- Il existe d'autres stratégies pour supprimer les sommets binaires, comme par exemple choisir le successeur/la feuille avec l'étiquette minimale du sous-arbre de droite.