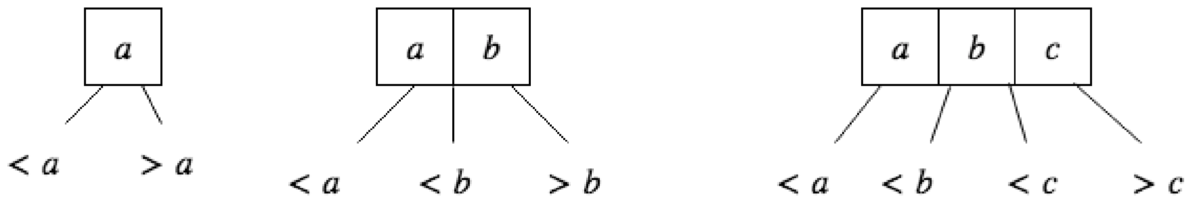
# Arbres 2-3-4

## Introduction[1]

Un arbre 2-3-4 est un arbre de recherche équilibré comportant trois les types de nœuds suivants.

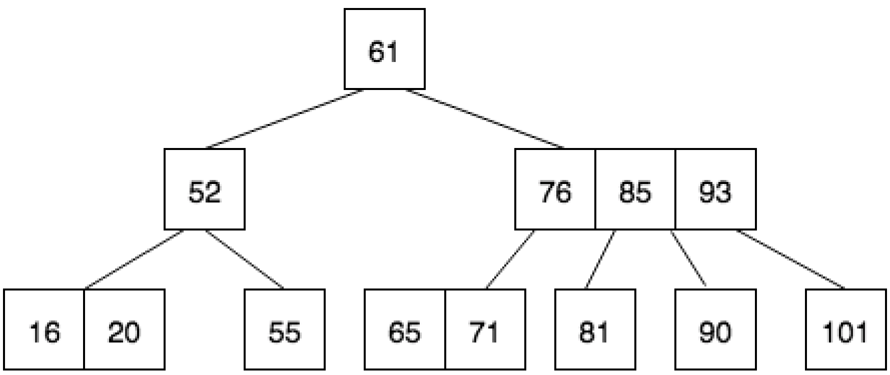
1. **2-noeud** a **une** clé et **deux** nœuds enfants (tout comme le nœud d'un ABR).
2. **3-nœud** a **deux** clés et **trois** nœuds enfants.
3. **4-nœud** a **trois** clés et **quatre** nœuds enfants.

La raison de l'existence de ces trois types  de noeuds est de rendre l'arbre parfaitement équilibré (tous les nœuds feuilles sont au même niveau) après chaque opération d'insertion et de suppression. La figure ci-dessous illustre graphiquement ces types de nœuds.




(a) 2-node (b) 3-node (c) 4-node

Si un nœud a plus d'une clé (3-nœud et 4-nœud), les clés doivent être dans l'ordre trié. Cela garantit que le parcours dans l'ordre donne toujours les clés dans l'ordre trié. Un exemple d'arbre 2-3-4 est donné dans la figure ci-dessous.




## Opérations sur les arbres 2-3-4

Nous discutons de trois opérations majeures sur l'arbre 2-3-4. La première et la plus simple est la recherche, la suivante est l'insertion et la dernière est la suppression. Toutes les opérations prennent un temps $O(log\ n)$ car la hauteur de l'arbre est dans l'ordre logarithmique.

[1] https://algorithmtutor.com/Data-Structures/Tree/2-3-4-Trees/

## Opération de recherche

Comparez l'élément à rechercher avec les clés du nœud et déplacez-vous dans la direction appropriée. Contrairement à un ABR où nous nous déplaçons soit vers l'enfant gauche, soit vers l'enfant droit, nous devons faire un choix entre trois ou quatre chemins différents. Comme les clés sont triées, il est évident de choisir le chemin en suivant la règle donnée dans la première figure.

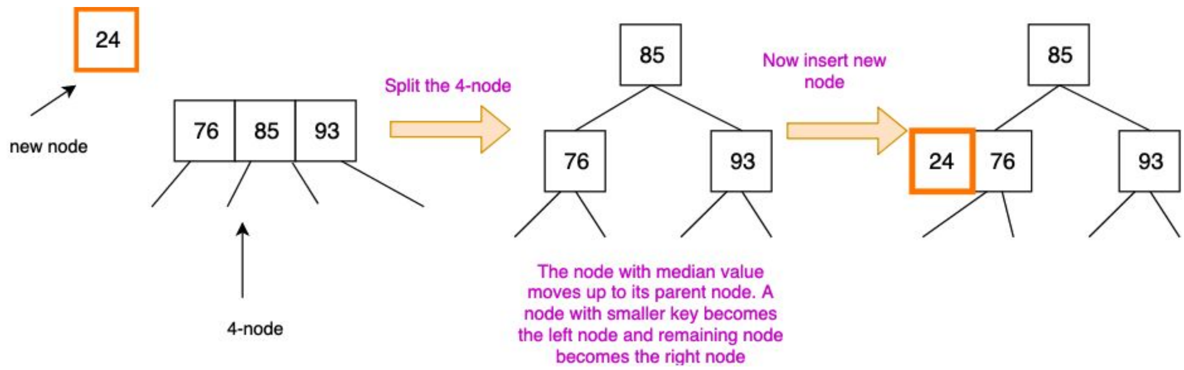
## Opération d'insertion

Un nœud ne peut pas contenir plus de trois clés. Si un nœud est plein avant l'insertion, on divise le nœud afin que le nouveau nœud puisse être inséré.

L'insertion a toujours lieu sur les nœuds feuilles. ***On n'insère jamais de nouveau nœud sur les nœuds internes même s'ils ont de la place pour les accueillir***. Par conséquent, on effectue l'opération de recherche sur l'arbre jusqu'à ce qu'on atteigne le nœud feuille.

Si le nœud feuille est un 2-nœud, nous insérons l'élément et en faisons un 3-nœud. Si le nœud feuille est un nœud à 3-nœuds, nous en faisons un 4-nœud.

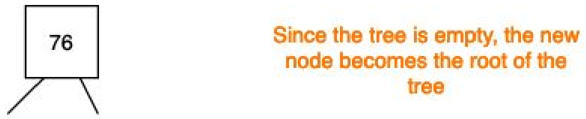
Mais que se passe-t-il si le nœud est un 4-nœud? On ne peut pas insérer un nouveau nœud dans ce nœud non? Le nœud est déjà plein. Dans ce cas, on divise le nœud en nœuds avec un plus petit nombre de clés et insère le nouveau nœud dans le nœud enfant approprié. Ceci est illustré dans la figure ci-dessous.



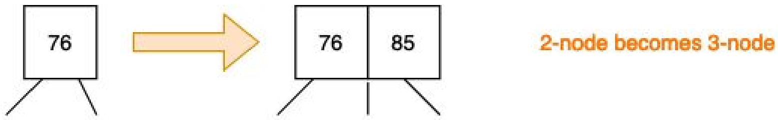

Le processus de fractionnement peut, parfois, aller jusqu'au nœud racine. Cela se produit lorsque tous les nœuds sur le chemin allant de la racine au nœud feuille sont à 4 nœuds. Lorsque le nœud racine se divise, il augmente la hauteur de l'arbre de 1.

La figure ci-dessous illustre l'opération d'insertion.

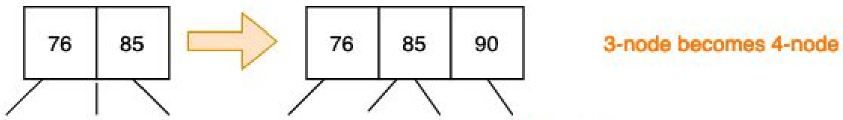
**Insert 76**



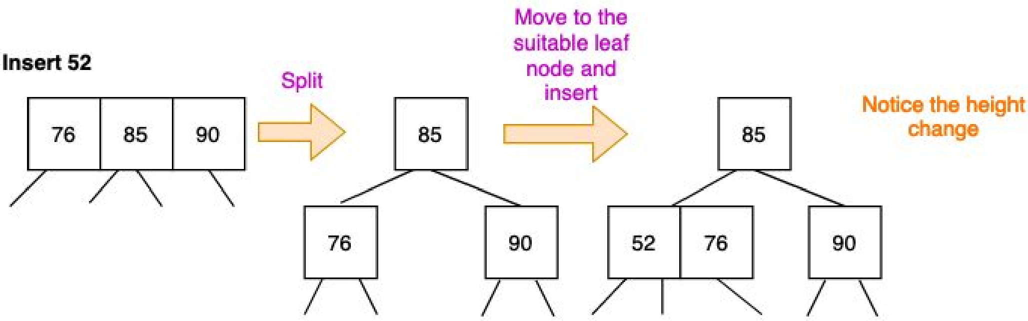

**Insert 85**



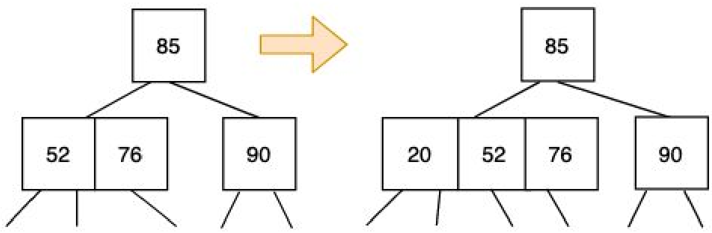

**Insert 90**



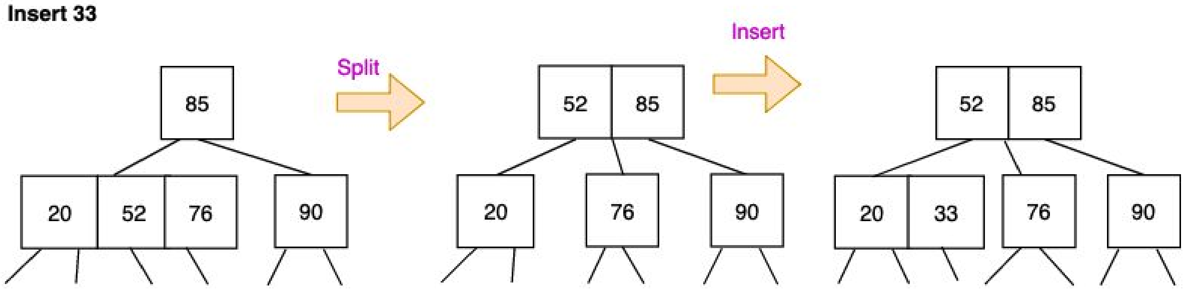

**Insert 52**




**Insert 20**




**Insert 33**



## Opération de suppression

La suppression est un peu plus compliquée que l'opération d'insertion. En fonction de l'emplacement du nœud contenant la cible X à supprimer, nous devons considérer plusieurs cas.

**Cas 1:** si X est dans un nœud feuille

Cela a deux autres cas.

a. **Cas 1.1:** Si  X est soit dans un 3 nœuds ou 4 nœuds

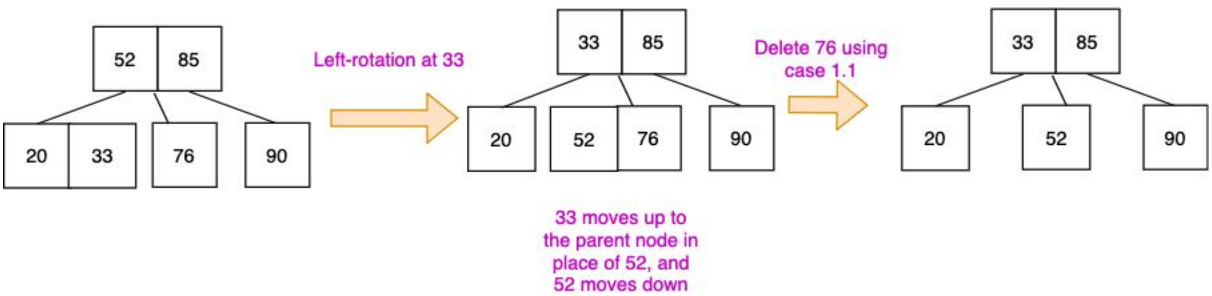
   Supprimer X. Si le nœud est un 3 nœuds, il devient 2 nœuds et si le nœud est un 4 nœuds, il devient 3 nœuds.

b. **Cas 1.2:** Si X est dans un 2 nœuds

   C'est ce qu'on appelle le *sous-débit* . Pour résoudre ce problème, nous devons examiner trois autres cas.

   i. **Cas 1.2.1:** Si le nœud contenant X a des frères et sœurs à 3 ou 4 nœuds

      Convertissez le 2-nœud en un 3-nœud en volant la clé du frère. Cela peut être fait par rotation gauche ou droite. Si le frère gauche est à 3 nœuds (ou 4 nœuds), faites la rotation à gauche, sinon faites la rotation à droite. La rotation à gauche est illustrée sur la figure ci-dessous. Ici, le nœud avec la clé 76 est en cours de suppression.




   La rotation droite est symétrique. Après la rotation, utilisez le cas 1.1 pour supprimer X.

   ii. **Cas 1.2.2:** Si les deux frères et sœurs sont à 2 nœuds mais que le nœud parent est soit à 3 nœuds, soit à 4 nœuds

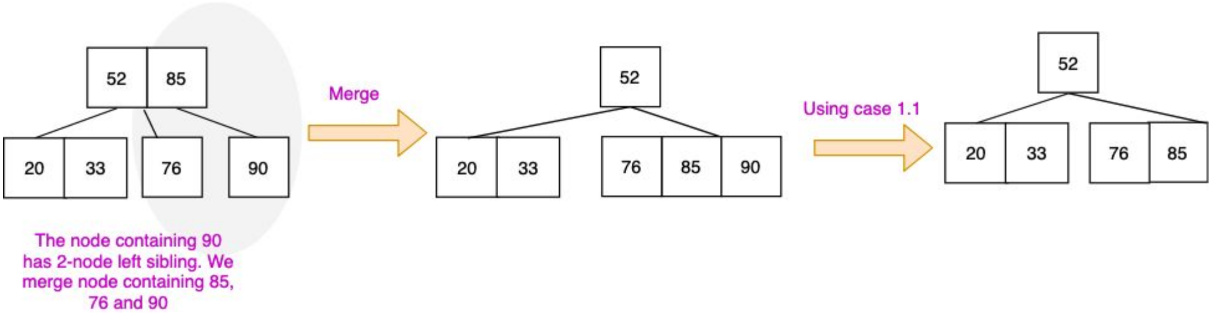
      Dans ce cas, nous convertissons le 2-nœud en un 4-nœud en utilisant l'opération de fusion (ou fusion). Nous fusionnons les trois nœuds suivants.

      Les 2 nœuds actuels contenant X.

Le nœud frère gauche ou droit (qui est également un nœud à 2).

Le nœud parent correspondant à ces deux nœuds.

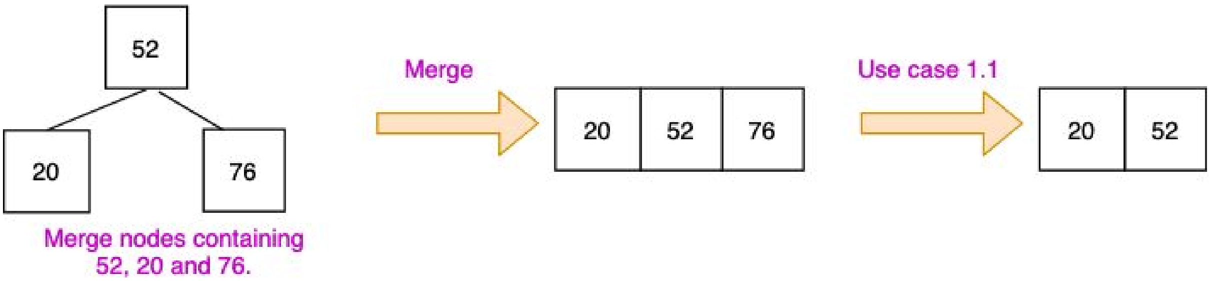
Après la fusion, les clés du nœud parent diminuent de 1. Ceci est illustré sur la figure ci-dessous où un nœud 2 contenant 90 est supprimé.




Après la fusion, nous utilisons le cas 1.1 pour supprimer X.

iii. **Cas 1.2.3:** Si les deux frères et le nœud parent sont un nœud à 2

Nous rencontrons rarement ce scénario. Dans ce cas particulier, le nœud parent doit être une racine. Tout comme la fusion, nous combinons les frères et sœurs et le nœud parent pour en faire un nœud à 4. Ce processus réduit la hauteur de l'arbre de 1. La figure ci-dessous illustre cela. Le nœud contenant 76 est en cours de suppression.




**Cas 2:** Si X est dans un nœud interne

Dans ce cas,

1. Trouvez le prédécesseur du nœud contenant X. Le prédécesseur est toujours un nœud feuille.
2. Échangez le nœud contenant X avec son nœud prédécesseur. Cela déplace le nœud contenant X au nœud feuille.
3. Puisque X est dans un nœud feuille, c'est le cas 1. Utilisez les règles données dans le cas 1 pour le supprimer.

# Tâches à Faire

En vous basant sur vos propres recherches et sur les ressources du cours, proposer un programmes avec les fonctions suivantes:

- insertion dans un arbre 2-3-4,
- suppression dans un arbre 2-3-4,
- recherche dans un arbre 2-3-4,
- transformation d'un arbre binaire de recherche (ABR) en un arbre 2-3-4,
- menu de choix pour effectuer les différentes opérations dans un arbre 2-3-4.

Afin de mieux tester l'utilité des arbres 2-3-4,

1. créer un ABR;
2. faire une fonction itérant sur l'insertion successive de 1000 à 10000 valeurs aléatoires dans l'ABR, avec une intervalle de 1000, c'est-à-dire, vous testez d'abord sur 1000, ensuite 2000, ensuite 3000, …, jusqu'à 10000 (remarque: vous pouvez aller aussi au delà, rien ne vous empêche de tester avec 100000 noeuds);
3. pour chaque test à la question précédente, appeler une fonction de recherche dans cet ABR, et estimer la durée de calcul;
4. comparer les durées en temps de calcul les différentes cardinalités (1000, 2000, 3000, 4000, 5000, …, 10000);
5. Qu'observez-vous ?
6. Refaire les étapes 2 à 4 cette fois ci avec un arbre 2-3-4.
7. Qu'observez-vous?
8. Comparer les durées obtenues avec les ABR et celles obtenues avec les arbres 2-3-4.
9. Quelles conclusions tirez-vous de ces étapes et comparaisons ?