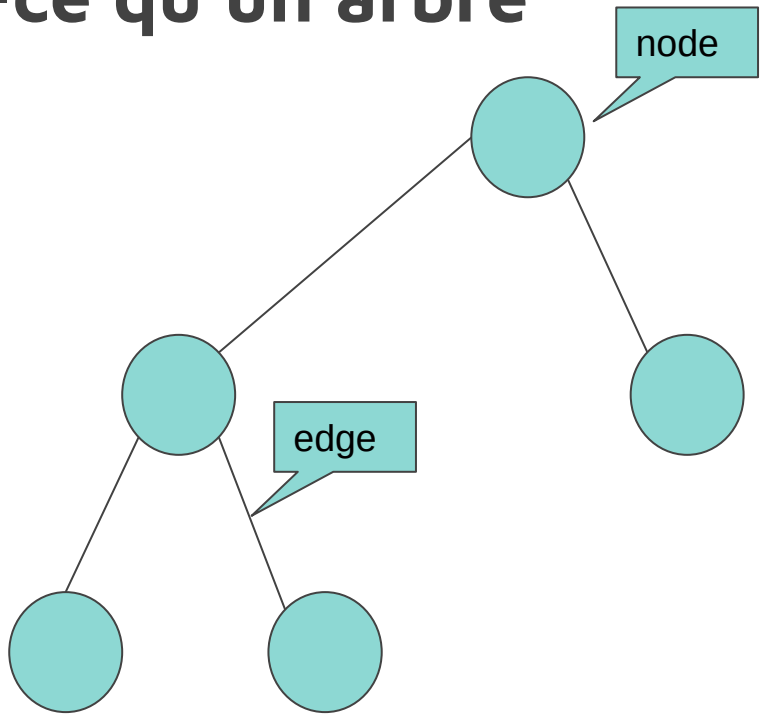
Programmation 3

Arbres binaires

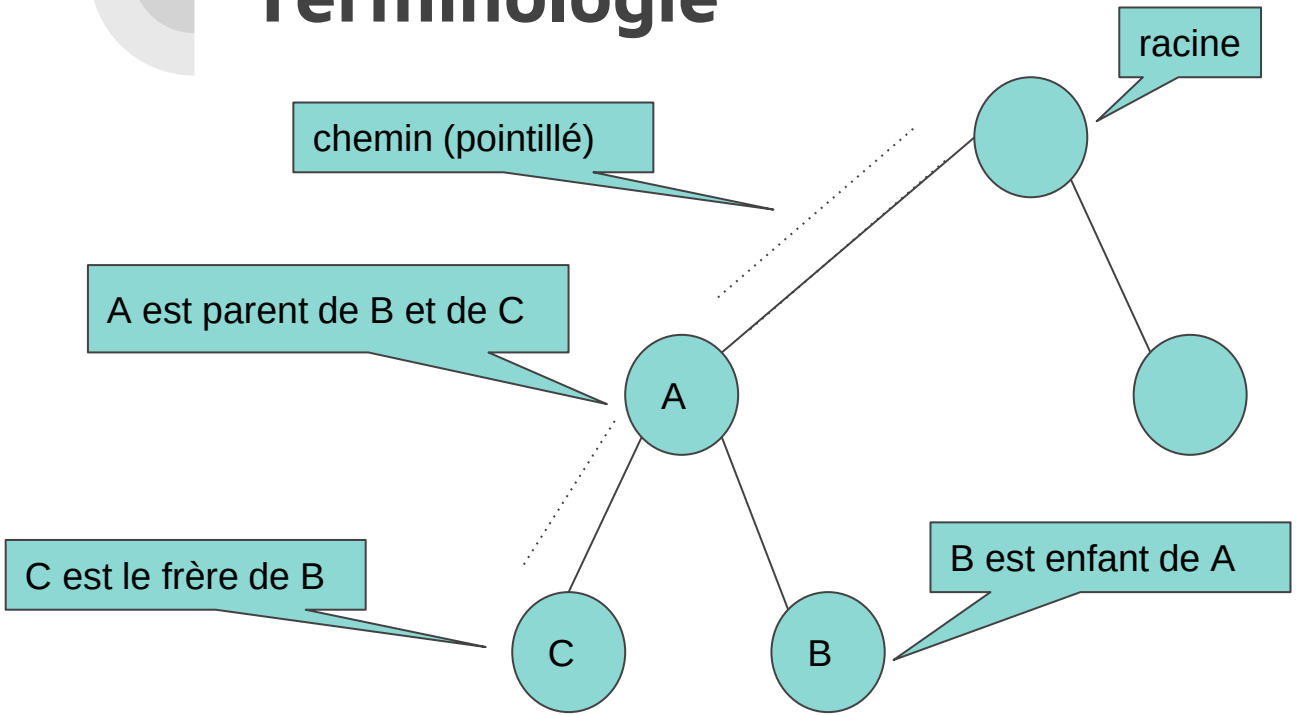
Qu’est-ce qu’un arbre



Pourquoi un arbre binaire

- Permet la recherche rapide comme un tableau trié
- Permet l’insertion rapide comme une liste liée

Terminologie



Règles d’un arbre binaire

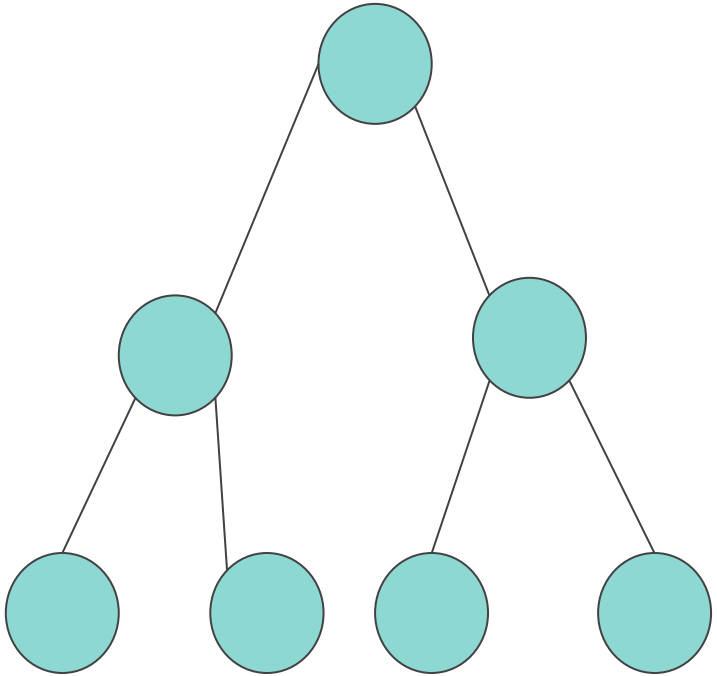
Un noeud (node) ne peut avoir que deux enfants

Les deux enfants se nomment “enfant de gauche” et “enfant de droite”

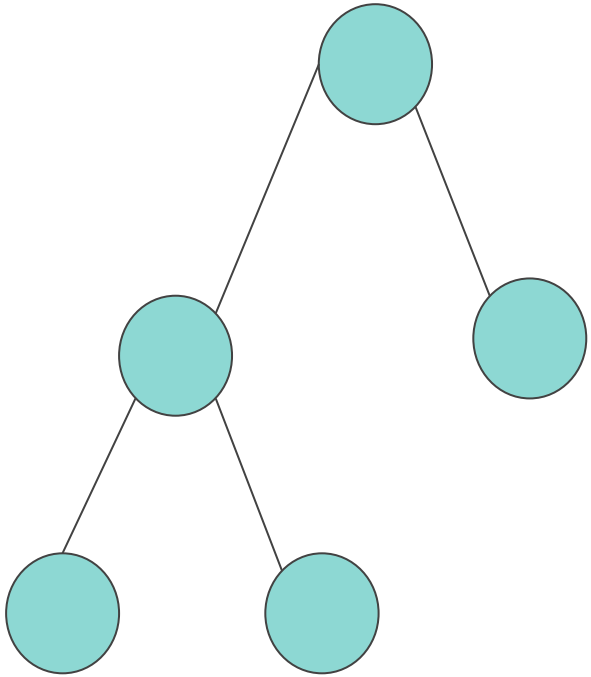
La clé de l’enfant de gauche doit être plus petite que la clé du parent.

La clé de l’enfant de droite doit être plus grande que la clé du parent.

Arbre balancé vs arbre non-balancé

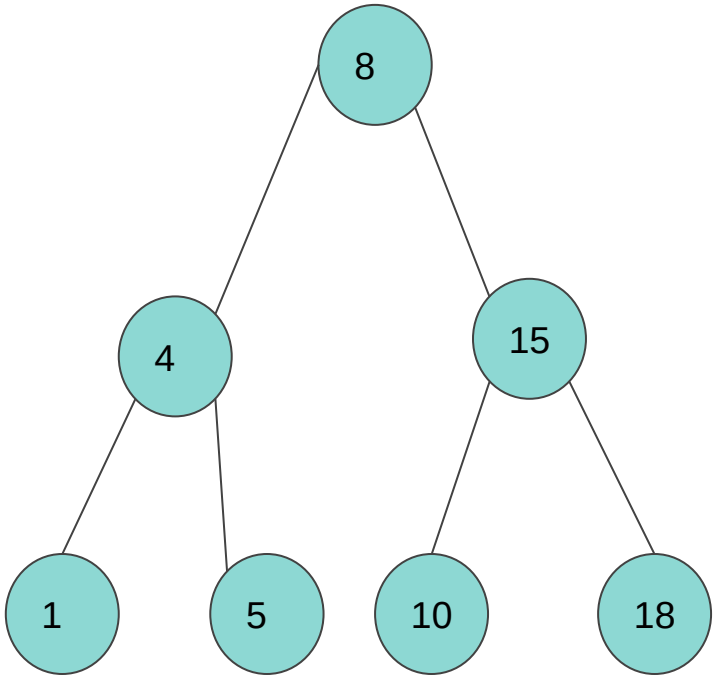


Balancé



Non-Balancé

Arbre balancé avec clés



Balancé

Une classe noeud

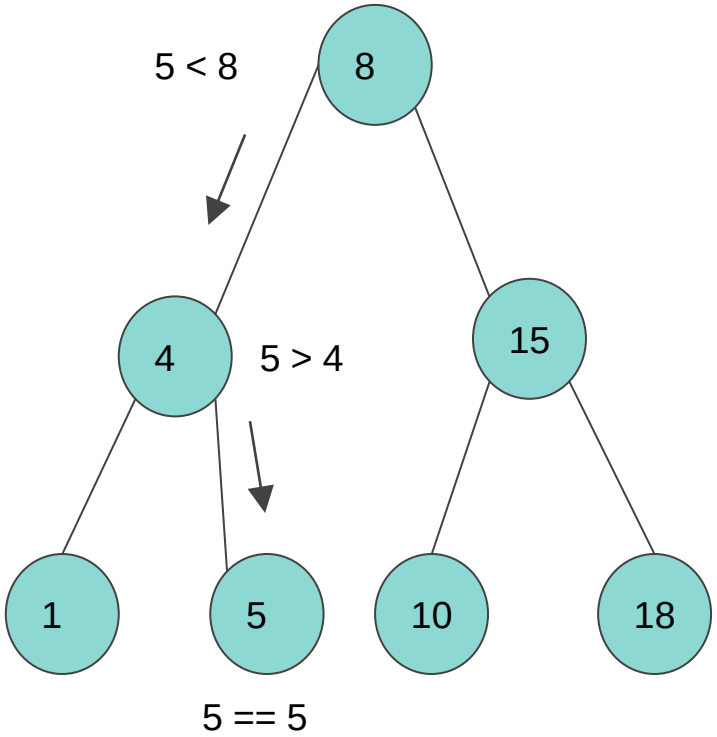
Une classe noeud a au minimum comme attributs:

- Sa clé
- Une référence à son enfant de gauche
- Une référence à son enfant de droite

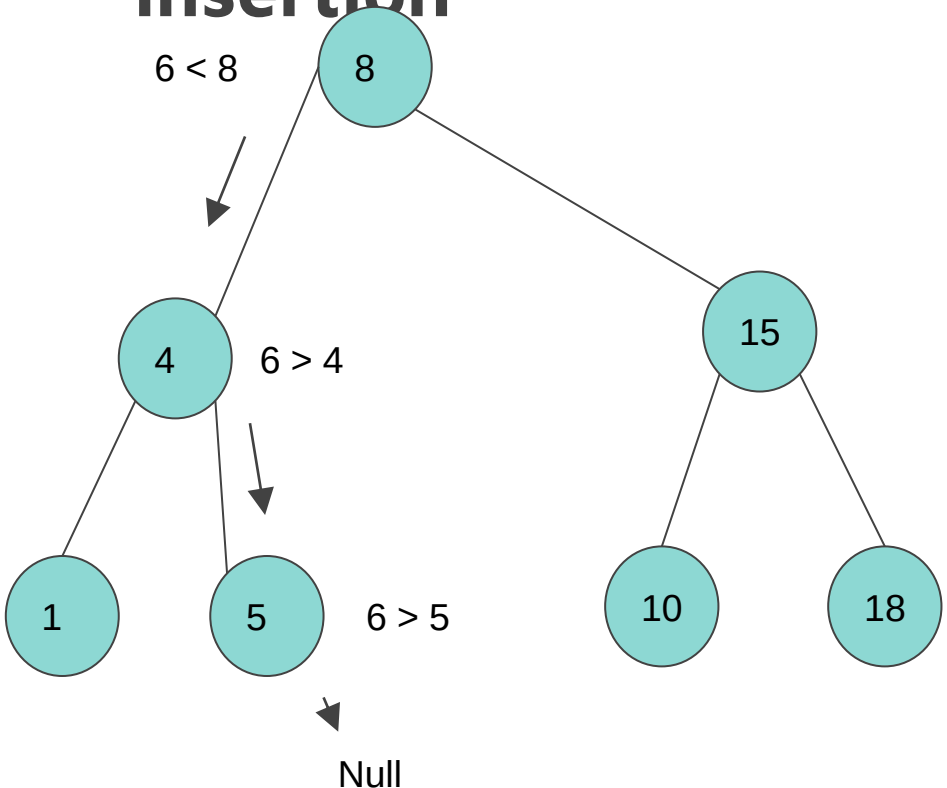
Ses méthodes obligatoires :

- Trouver
- Ajouter
- Supprimer

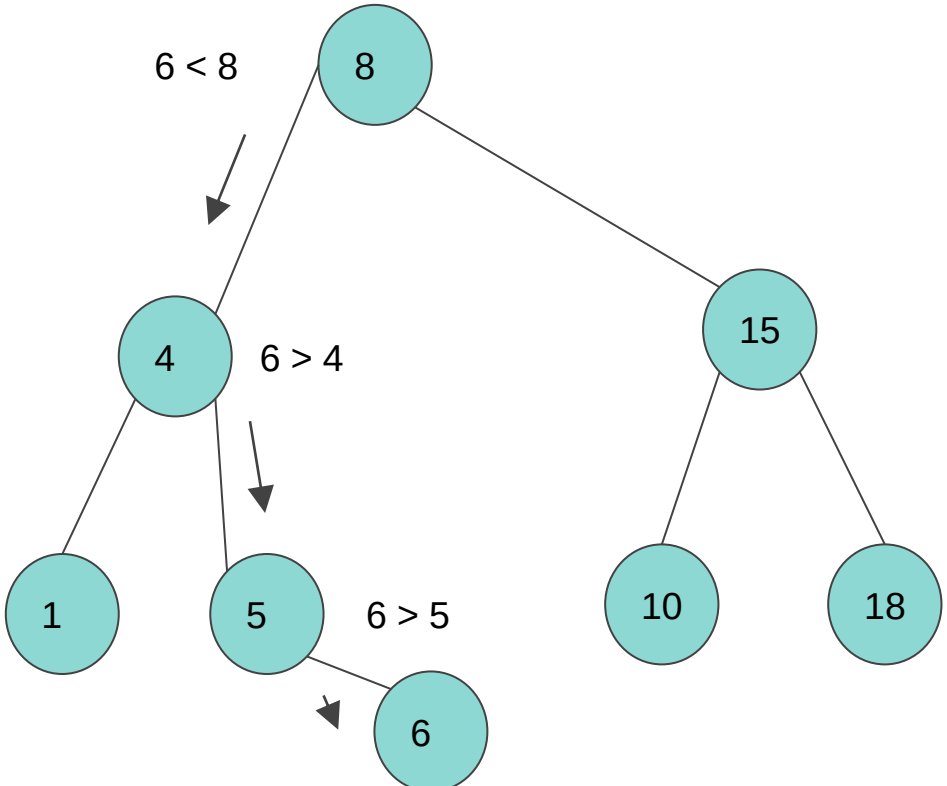
Trouver un noeud (ex: 5)



Ajouter un noeud (ex: 6) - Avant insertion



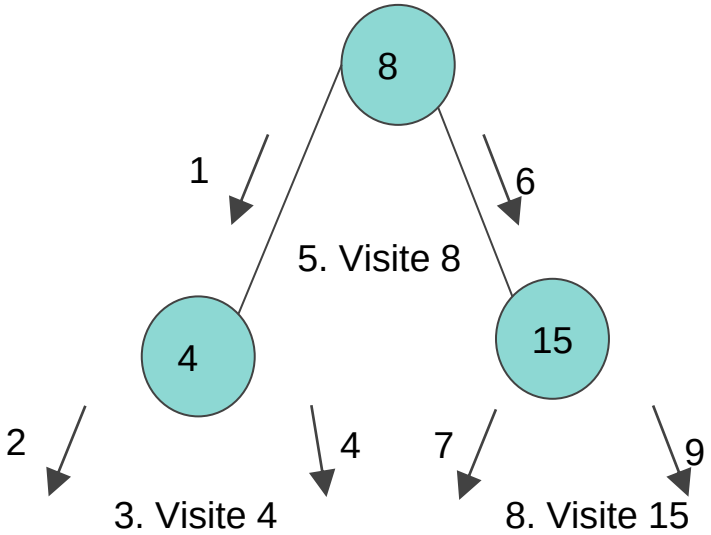
Ajouter un noeud (ex: 6) - Après insertion



Traverser un arbre binaire

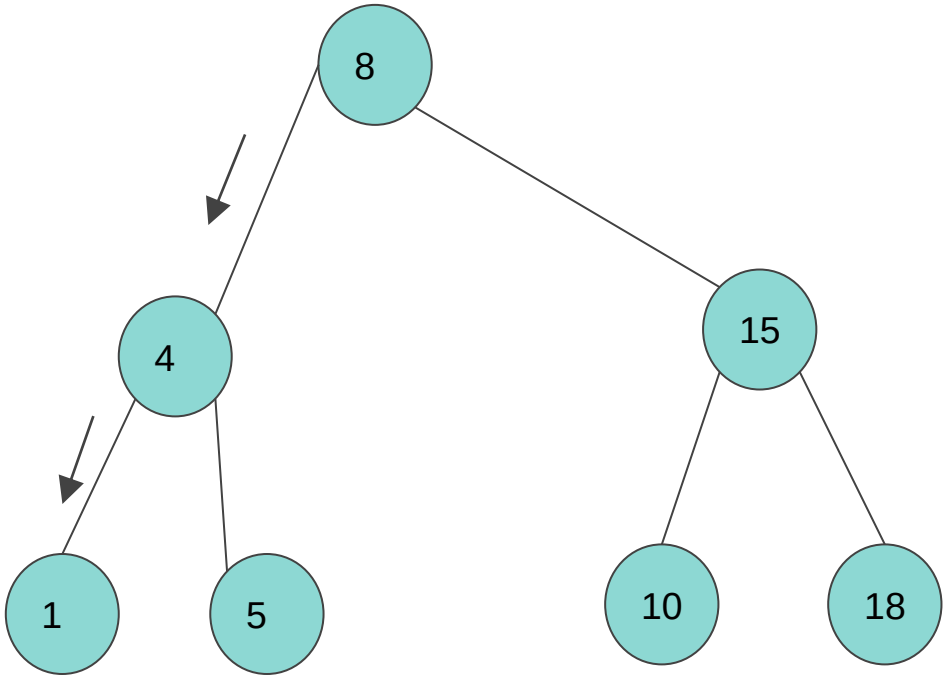
- Lister les entrées en ordre
- Peut retourner un tableau des éléments triés
- Utiliser la récursivité :
  - S'appeler pour lire son enfant de gauche
  - Visiter le noeud (par exemple afficher sa valeur)
  - S’appeler pour lire son enfant de droite

Traverser un arbre, exemple

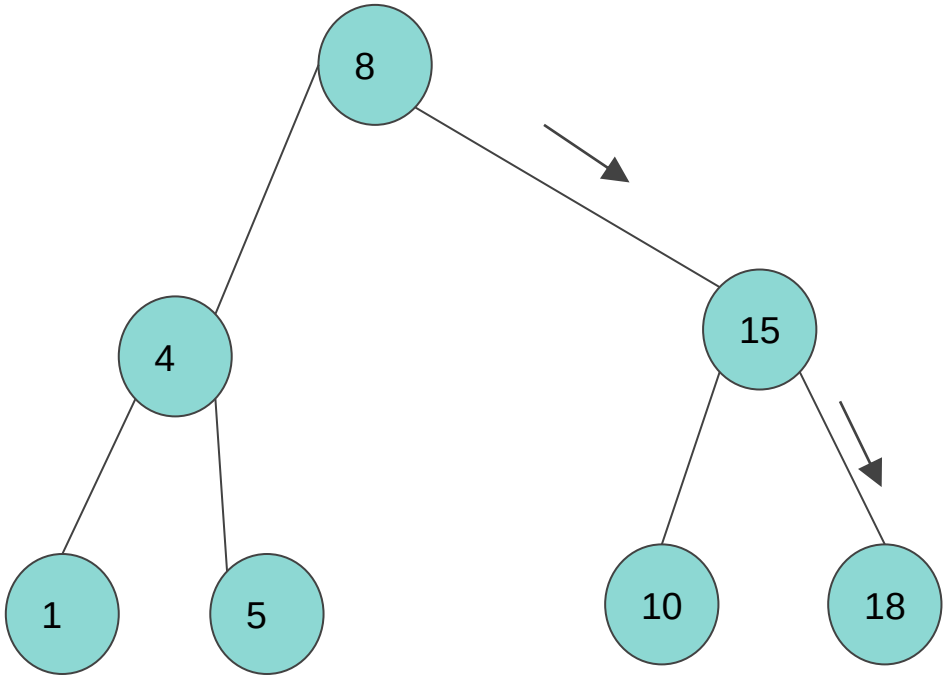


Résultat de la visite : 4 - 8 - 15

Trouver le minimum



Trouver le maximum



Exercice 1

- Faire une classe BinaryNode
  - Implémenter les méthodes :
    - add(key, value)
    - find(key)
    - traverse()
    - min()
    - max()