

# TP 3

## Structure de données

Shir-Li Kedem

Sciences informatiques

18/04/2023

### 2 - Phonebook du SRC

#### 2.2 Modélisation

1. La structure de données utilisée ici est l'arbre binaire, qui permet de trier chaque personne de manière organisée. composée de nœuds reliés entre eux de manière hiérarchique. Le nœud situé tout en haut de l'arbre est appelé la racine (root). Chaque nœud peut avoir jusqu'à deux nœuds enfants, qui sont appelés le nœud gauche et le nœud droit. Comme l'arbre binaire a une complexité de recherche en  $O(\log n)$ , c'est une solution efficace pour effectuer des recherches.

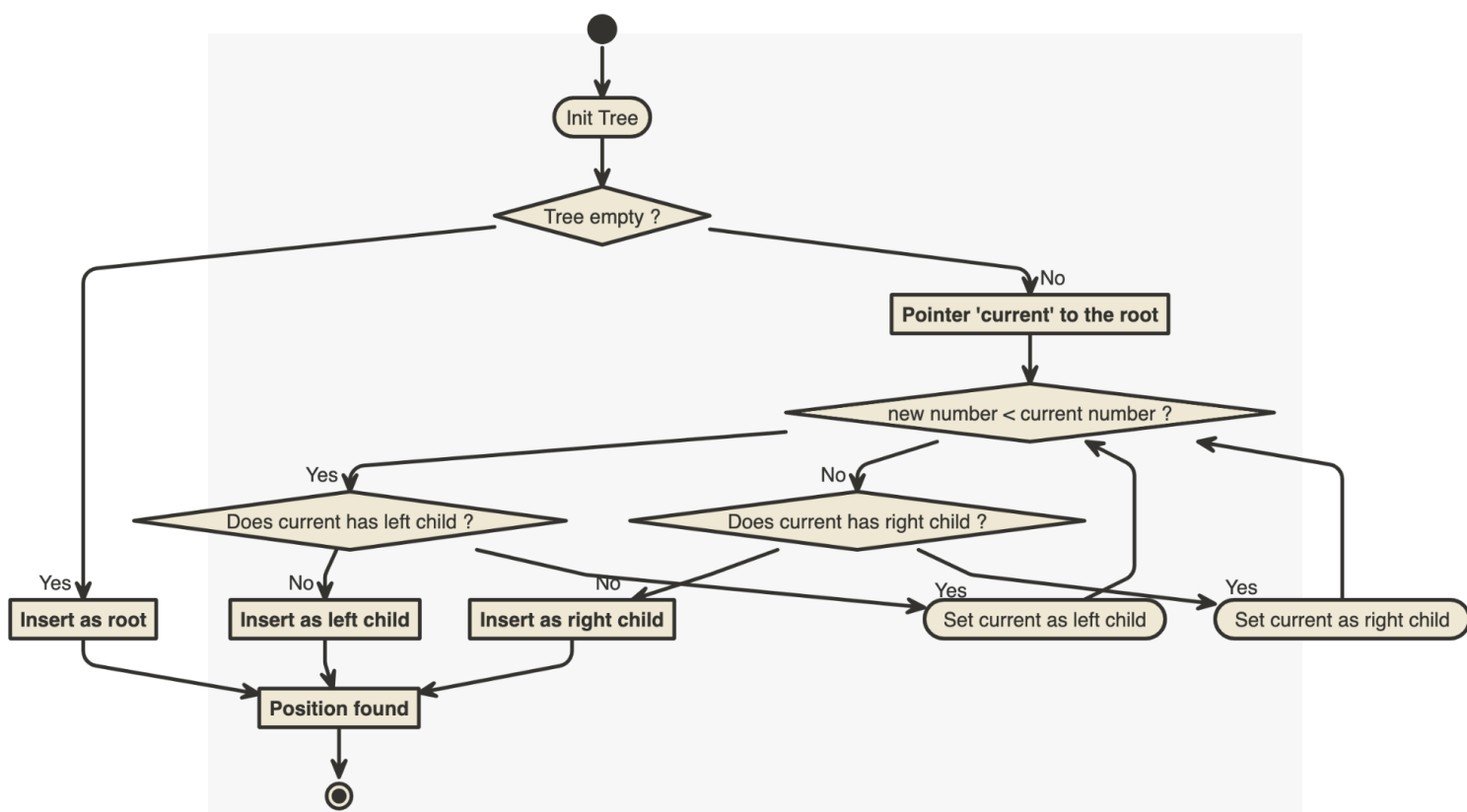
Dans un arbre binaire (TTree), chaque nœud est étiqueté avec une valeur, qui peut être une donnée quelconque, ici chaque nœud représente une personne (TPerson). Les nœuds enfants sont organisés de sorte que la valeur étiquetée dans le nœud gauche est inférieure à la valeur étiquetée dans le nœud parent, et que la valeur étiquetée dans le nœud droit est supérieure à la valeur étiquetée dans le nœud parent. Ici, cette valeur est le numéro de téléphone (TNumber).

Voici les représentations TAD pour TTree, TPerson et TNumber.

TTree	TPerson
Create : a) Initialization - TTree * init() b) Create elements with CSV file - TTree * populate()	Create : a) Create new person - TPerson * create()
Update : a) Add element to the tree - void insert()	Read : a) Find person informations by phone number - TPerson * find() b) Show person's informations - void print()

TNumber
Update : a) String to int phone number - TNumber convert()

2. Une clé de tri est une fonction ou un critère qui est utilisé pour ordonner une collection d'éléments selon un ordre donné. Elle peut être utilisée pour trier une liste d'objets en fonction d'une propriété particulière de chaque objet, ou pour trier une liste de valeurs en fonction de leur relation les unes avec les autres. Pour cet exercice, nous utiliserons le numéro de téléphone comme clé de tri afin de répondre aux besoins du SRC.
3. Voici ci-dessous le diagramme d'activité de la procédure utilisée pour indexer les personnes dans l'arbre. En fait, cette procédure est la fonction "insert()" qui est détaillée dans l'implémentation.



4. Une autre clé que nous pourrions utiliser est la clé nom-prénom. Cette clé est toute aussi pertinente car avec l'ordre du nom-prénom (ordre alphabétique) on obtient directement des sous-arbres qui sont en fait les personnes faisant partie de la même famille, ce qui peut faciliter le réseau de recherche en cas d'indisponibilité de la personne désirée.

## 2.3 Implémentation

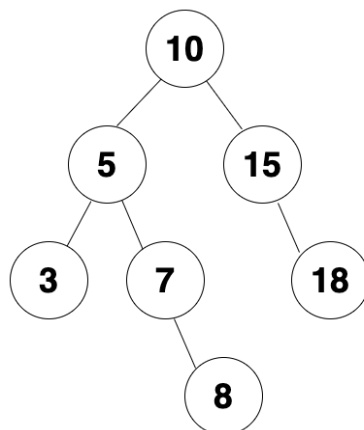
1. Voir dans le code.

2. Après quelques recherches de numéros de téléphones aléatoires avec le programme implémenté, on remarque que le temps écoulé pour trouver la réponse est très rapide par rapport à une recherche linéaire classique. En effet, si on regarde plus précisément, avec la recherche linéaire on obtient une complexité en temps de  $O(N)$ , où  $N$  est le nombre de personnes dans la base de données. Tandis qu'avec notre programme implémenté, comme il s'agit d'un arbre binaire, la complexité en temps est de  $O(\log N) < O(N)$ , ce qui explique la rapidité du programme.

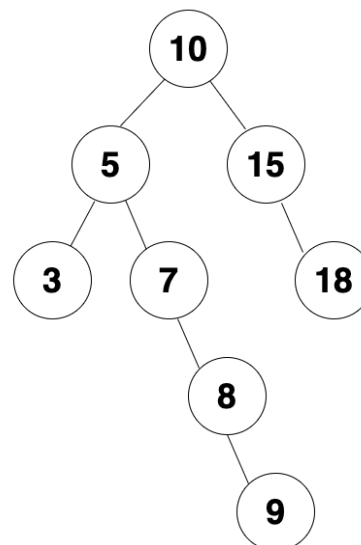
### 3 - Arbres AVL

1. Un arbre AVL est une structure de données d'arbre binaire de recherche équilibré par hauteur. Il maintient l'équilibre en réorganisant l'arbre à l'aide d'une technique de rotation lorsqu'un nœud est inséré ou supprimé. Cela garantit une efficacité de recherche en temps logarithmique.

2.



Arbre AVL



Arbre Non AVL

Le premier arbre à gauche de la figure est AVL car chaque sous-arbre gauche et droit d'un nœud a une hauteur qui ne diffère jamais de plus d'une unité.

Le second arbre à droite de la figure n'est pas AVL car la différence de hauteur entre le sous-arbre gauche et le sous-arbre droit du nœud 5

est de 2 unités, ce qui ne respecte pas la règle d'équilibre d'un arbre AVL.

### 3. Propriétés des arbres AVL :

- Les sous-arbres gauche et droit de chaque nœud ont des hauteurs qui diffèrent d'au plus une unité.
- Tous les sous-arbres d'un arbre AVL sont eux-mêmes des arbres AVL.
- La hauteur d'un arbre AVL avec  $n$  nœuds est  $O(\log n)$ .
- L'opération de recherche, insertion et suppression a une complexité en temps de  $O(\log n)$ .

### Complexités des opérations sur un arbre AVL :

- Recherche :  $O(\log n)$
- Insertion :  $O(\log n)$
- Suppression :  $O(\log n)$

4. Le rééquilibrage est un processus pour maintenir l'équilibre d'un arbre en ajustant les nœuds de l'arbre. Dans le cas d'un arbre AVL, le rééquilibrage est effectué par une technique de rotation lorsqu'un nœud est inséré ou supprimé. Le but étant de garantir que les sous-arbres gauche et droit de chaque nœud ont des hauteurs qui ne diffèrent pas de plus d'une unité. Ainsi, le rééquilibrage est important et nécessaire pour garantir une performance efficace des opérations de l'arbre.
5. Le facteur d'équilibre est une mesure de l'équilibre d'un nœud dans un arbre AVL. Il est défini comme la différence de hauteur entre le sous-arbre gauche et le sous-arbre droit d'un nœud. Le facteur d'équilibre peut prendre trois valeurs : -1, 0 ou 1. Si ce facteur est en dehors de cette intervalle, l'arbre n'est pas équilibré.
6. - Rotation simple droite : Utilisé lorsque le facteur d'équilibre d'un nœud est -2, ce qui signifie que le sous-arbre gauche est plus grand que le sous arbre droit de 2 niveaux.  
→ Elle consiste à déplacer le sous-arbre gauche du nœud pivot vers la droite, en faisant du sous-arbre gauche le nouveau sous-arbre droit du nœud pivot.

- Rotation simple gauche : Utilisé lorsque le facteur d'équilibre d'un nœud est 2, c'est le contraire de la rotation simple, c'est-à-dire que le sous-arbre droit est plus grand que le sous-arbre gauche de 2 niveaux.

→ Elle consiste à déplacer le sous-arbre droit du nœud pivot vers la gauche, en faisant du sous-arbre droit le nouveau sous-arbre gauche du nœud pivot.

- Rotation double gauche droite : Utilisé lorsque le facteur d'équilibre d'un nœud est de -2 et que le facteur d'équilibre de son sous-arbre droit est 1, ce qui signifie qu'il y a un déséquilibre à gauche-droite.

→ Cette rotation consiste en une rotation simple gauche suivie d'une rotation simple droite.

- Rotation double droite gauche : Utilisé lorsque le facteur d'équilibre d'un nœud est de 2 et que le facteur d'équilibre de son sous-arbre gauche est de -1, ce qui signifie qu'il y a un déséquilibre à droite-gauche.

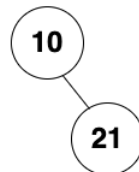
→ Cette rotation consiste en une rotation simple droite suivie d'une rotation simple gauche.

## 7. Voici les opérations

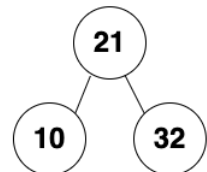
Insertion de l'élément 10  
Aucune rotation



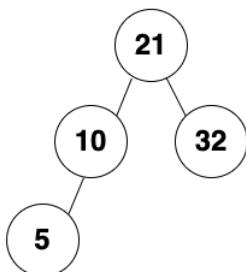
Insertion de l'élément 21  
Aucune rotation



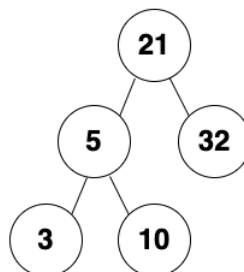
Insertion de l'élément 32  
Rotation simple gauche



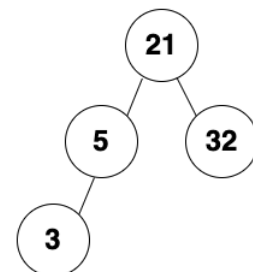
Insertion de l'élément 5  
Aucune rotation



Insertion de l'élément 3  
Rotation simple droite



Suppression de l'élément 10  
Aucune rotation



Suppression de l'élément 3  
Aucune rotation

