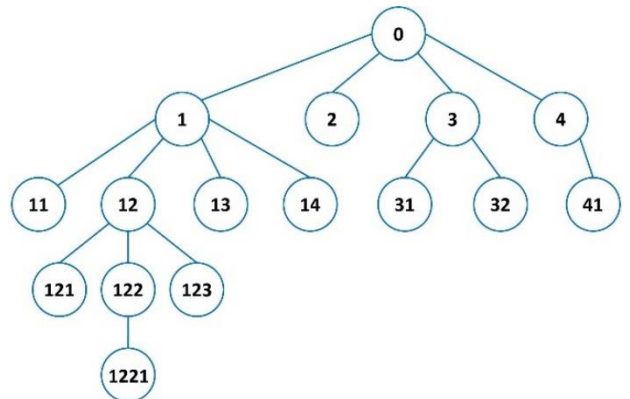


### Objectif

- Maîtriser le concept d'**arbre**.
- Implémenter un arbre **n-aire** (arbre avec plusieurs fils).
- Définir des classes utilisant un **type générique**.

### Énoncé du problème

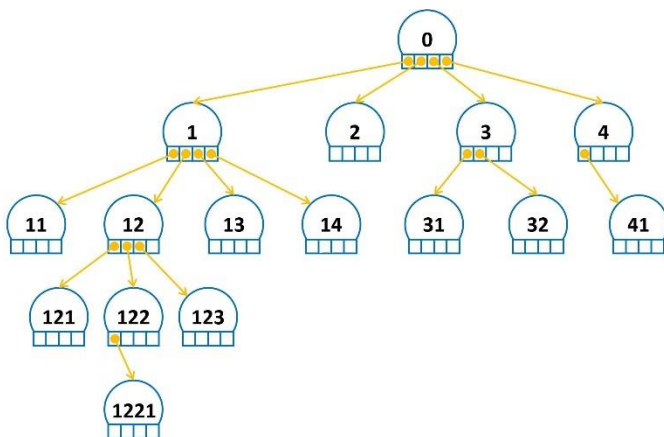
On vous demande de fournir une classe qui stocke des éléments sous forme arborescente. Chaque élément est rattaché à un seul père (à l'exception de la racine), et peut avoir zéro, un ou plusieurs fils.



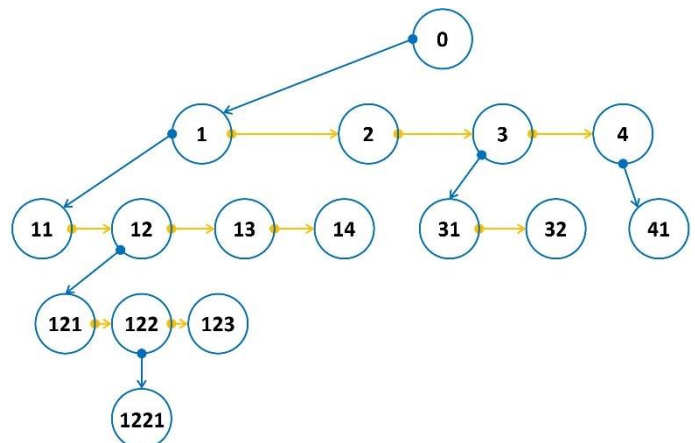
### Différentes implémentations

Deux implémentations différentes peuvent facilement être envisagées : un tableau des fils, ou une liste des frères. Comme d'habitude, l'implémentation la plus judicieuse dépend du contexte, du nombre potentiel de fils, des fonctionnalités principales, de l'utilisation, ...

#### Tableau (ou ArrayList, ou liste, ...) des fils



#### Fils (aîné ou cadet) et liste des frères



### Votre travail :

- Créez une classe `ArbreN` (avec l'implémentation que vous souhaitez) et implémentez les 3 méthodes suivantes :
- `insérer(valeur, père)` : qui insère un nouveau fils rattaché au père spécifié
- `chercher(valeur)` : qui retourne le nœud demandé
- `parcourir()` : qui parcourt tout l'arbre

### Remarques :

- Il n'y a pas de contraintes sur le contenu (ce n'est pas un arbre de recherche, avec un ordre !)
- Lors de l'insertion d'un nouveau nœud, il faut donc fournir la référence de son père.