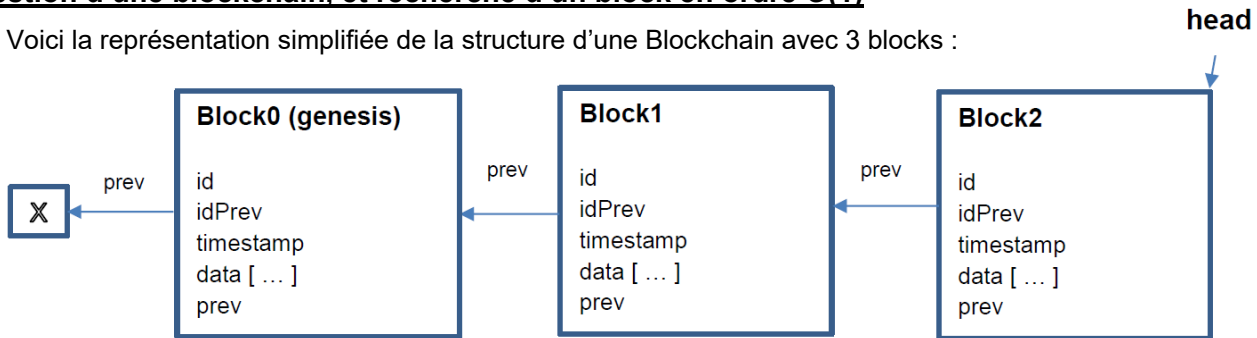


**Objectif**

- Utiliser une **liste chaînée**.
- Rechercher un élément dans une **liste chaînée**.
- Récupérer un élément dans une **liste chaînée** en ordre constant **O(1)**.

**Gestion d'une blockchain, et recherche d'un block en ordre O(1)**

- Voici la représentation simplifiée de la structure d'une Blockchain avec 3 blocks :



- **Énoncé** : construire une Blockchain contenant les blocks fournis :
  - les Block contiennent déjà les données (data) et le timestamp
  - initialiser les autres attributs, et conserver une référence sur la blockchain (head)
  - rechercher un block (search) et afficher cette Blockchain (toString)
- **Travail à faire** : compléter le **constructeur** de la blockchain :
  - le block reçu contient déjà le timestamp et les data, prev est à null
  - définir l'id
  - mettre à jour head
- Compléter la méthode **insert** qui reçoit un id et un block :
  - vérifier que l'id reçu est le même que l'id de head
  - si pas le même, lancer InvalidIdException
  - définir les attributs manquants dans le block (id, idPrev, prev)
  - mettre à jour head
- Compléter la méthode **search** qui recherche un block de la chaîne.
  - objectif: retourner le block dont l'id est la valeur reçue en paramètre (sans utiliser la méthode get ci-dessous).
- Compléter la méthode **get** qui retourne un block de la chaîne en ordre constant **O(1)** :
  - objectif: retourner tousLesBlocks.get(i) en ordre constant O(1), c'est-à-dire **sans parcourir la chaîne** !
  - par conséquent, on a besoin d'avoir un accès direct à tousLesBlocks
  - déclarer un attribut tousLesBlocks, et le remplir avec chaque block
  - et donc ici, dans get ==> retourner tousLesBlocks.get(i) en ordre constant O(1)
- Compléter la méthode **toString** qui affiche l'id de chaque block de la blockchain :
  - version 1 : retourner un String contenant l'id de tous les blocks
  - version 2 : avoir une std permettant de les retourner dans l'ordre d'insertion (block0, block1, block2, ...)
- Compléter la méthode **main** qui crée une blockchain contenant tous les blocks, puis l'affiche :
  - créer une blockchain (avec le bloc 0)
  - insérer tous les blocks (1 à n-1) dans la blockchain
  - afficher la blockchain
  - rechercher un bloc en utilisant search et get.