

Liste chaînée



Objectifs

- Principes et structure des listes chaînées
- Quelques opérations

Liste chaînée

- C'est une autre façon de stocker des collections d'objets
- Cette structure de données utilise la notion de chaîne
 - Chaque valeur est stockée dans un nœud (maillon d'une chaîne)
 - Les nœuds sont mis les uns à la suite des autres (chaîne)
- La structure de données est récursive :
 - chaque nœud connaît le nœud suivant
 - dans le cas de la liste doublement chaînée, chaque nœud connaît aussi le nœud précédent
- Afin de simplifier la manipulation de la liste, on va stocker les informations de la chaîne dans une structure de données représentant la liste

Structures de données

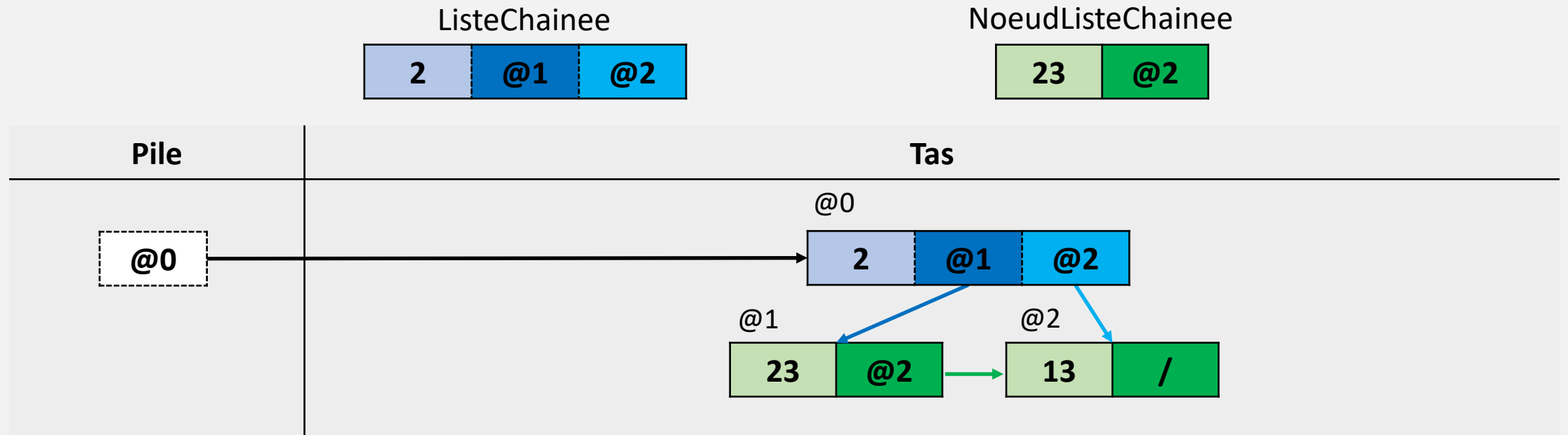
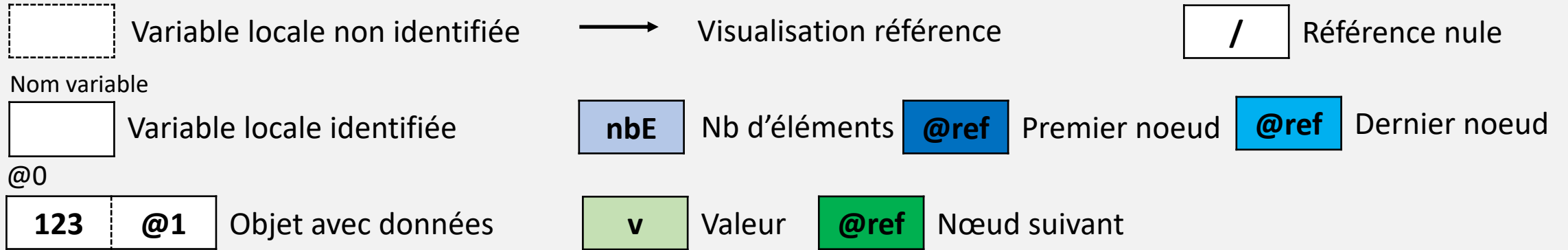
- Noeud :

- Valeur : valeur à stocker
 - Suivant : référence (lien) vers le nœud suivant, ou référence nulle si dernier de la liste
 - (Précédent) : référence (lien) vers le nœud précédent, ou référence nulle si premier de la liste
- Simplement chaînée
- Doublement chaînée

- Liste chaînée :

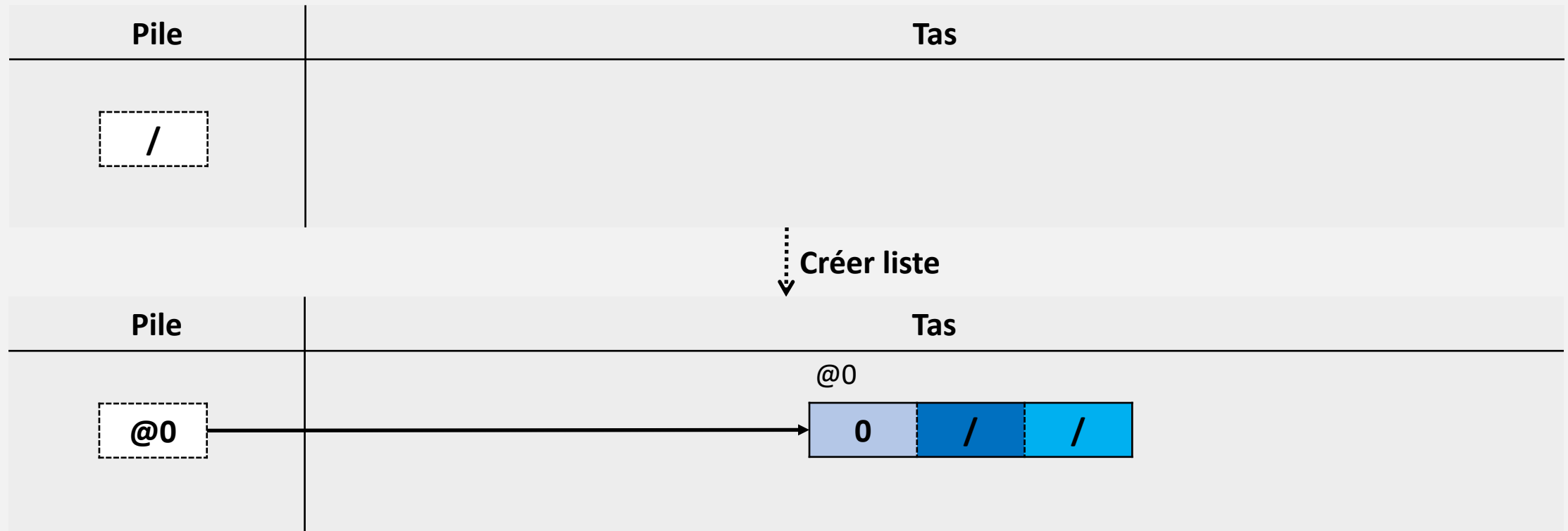
- Premier : référence (lien) vers le premier nœud de la chaîne, ou référence nulle si vide
 - Dernier : référence (lien) vers le dernier nœud de la chaîne, ou référence nulle si vide
 - Nombre d'éléments : nombre d'éléments contenus dans la liste
- Version optimisée pour certains algorithmes

Légendes et représentation graphique



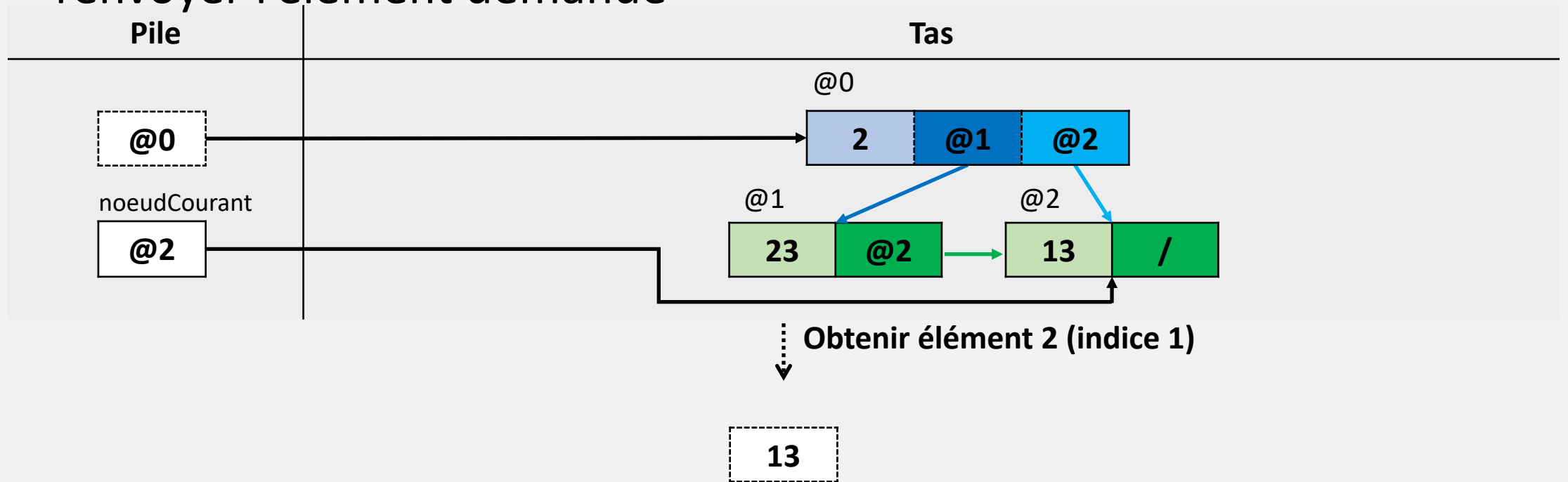
Opération – Créer une liste

- Idée : créer une liste chaînée avec chaque propriété à 0 ou référence nulle



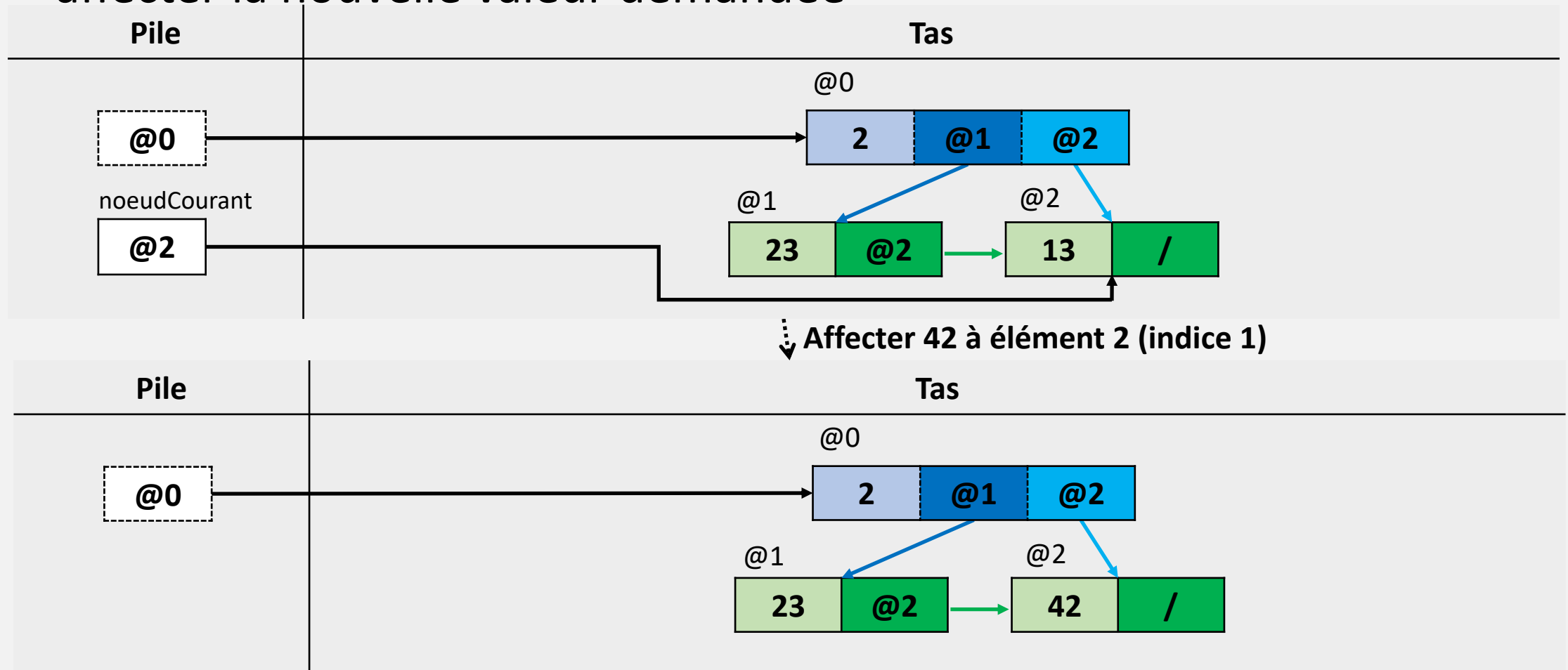
Opération – Accès à un élément – Lecture

- Idée : rechercher le nœud correspondant à l'indice demandé et renvoyer l'élément demandé



Opération – Accès à un élément – Écriture

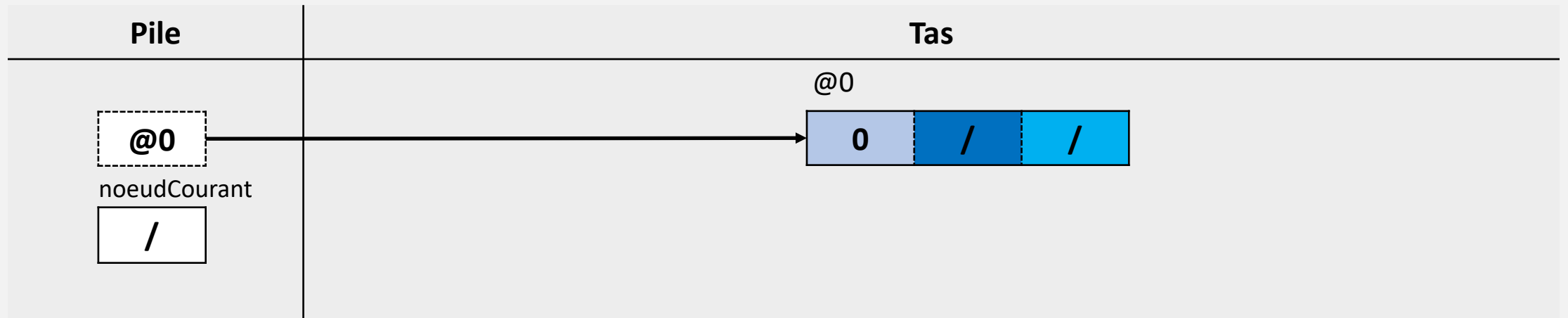
- Idée : rechercher le nœud correspondant à l'indice demandé et affecter la nouvelle valeur demandée



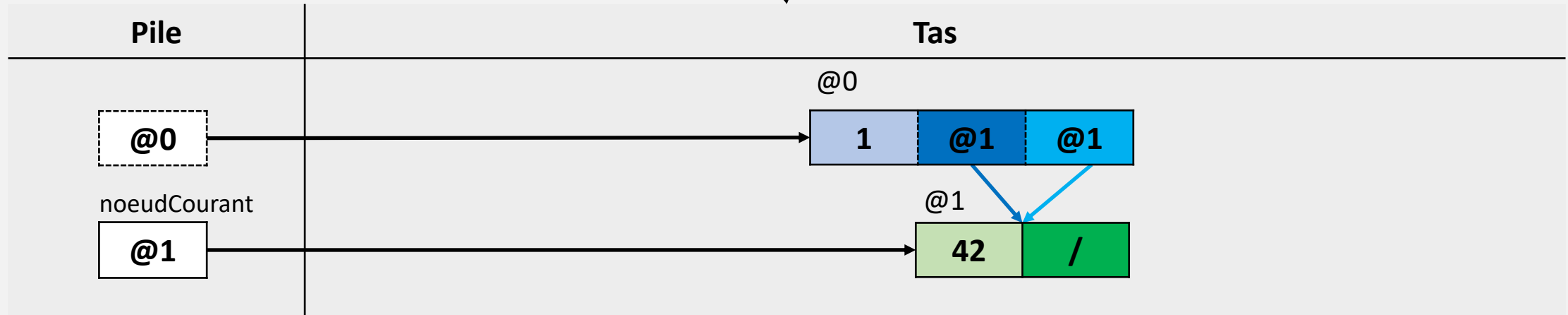
Opération – Ajout en début / fin de liste

Cas liste vide

- Idée

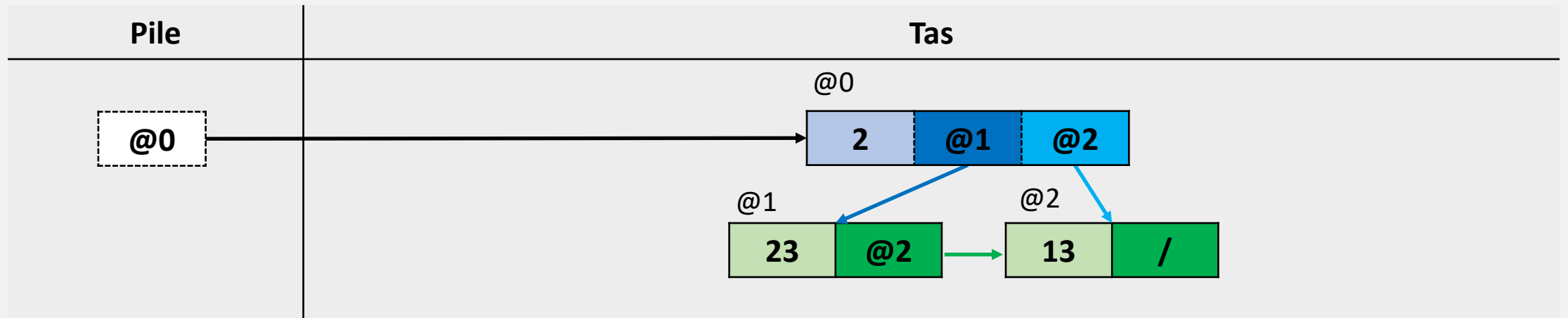


⇓ Ajouter 42 en début de liste

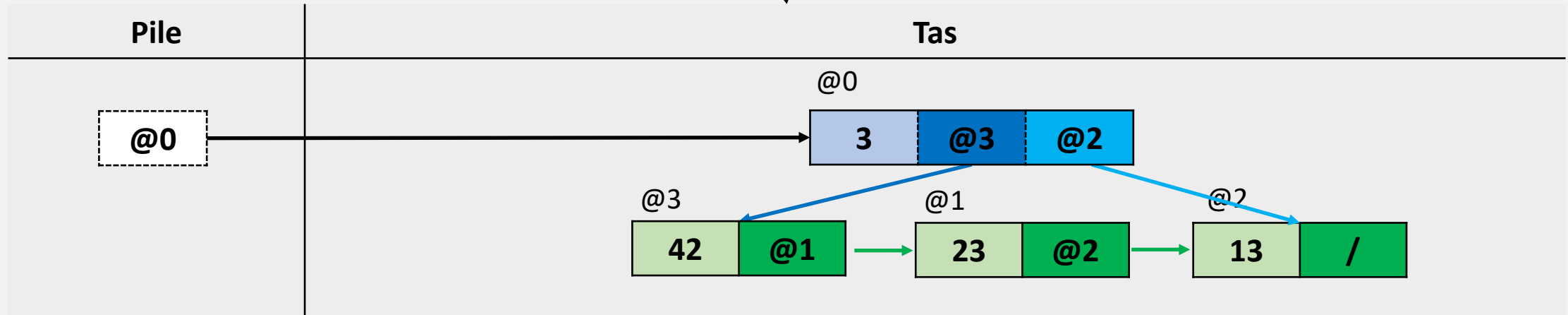


Opération – Ajout en début de liste

- Idée

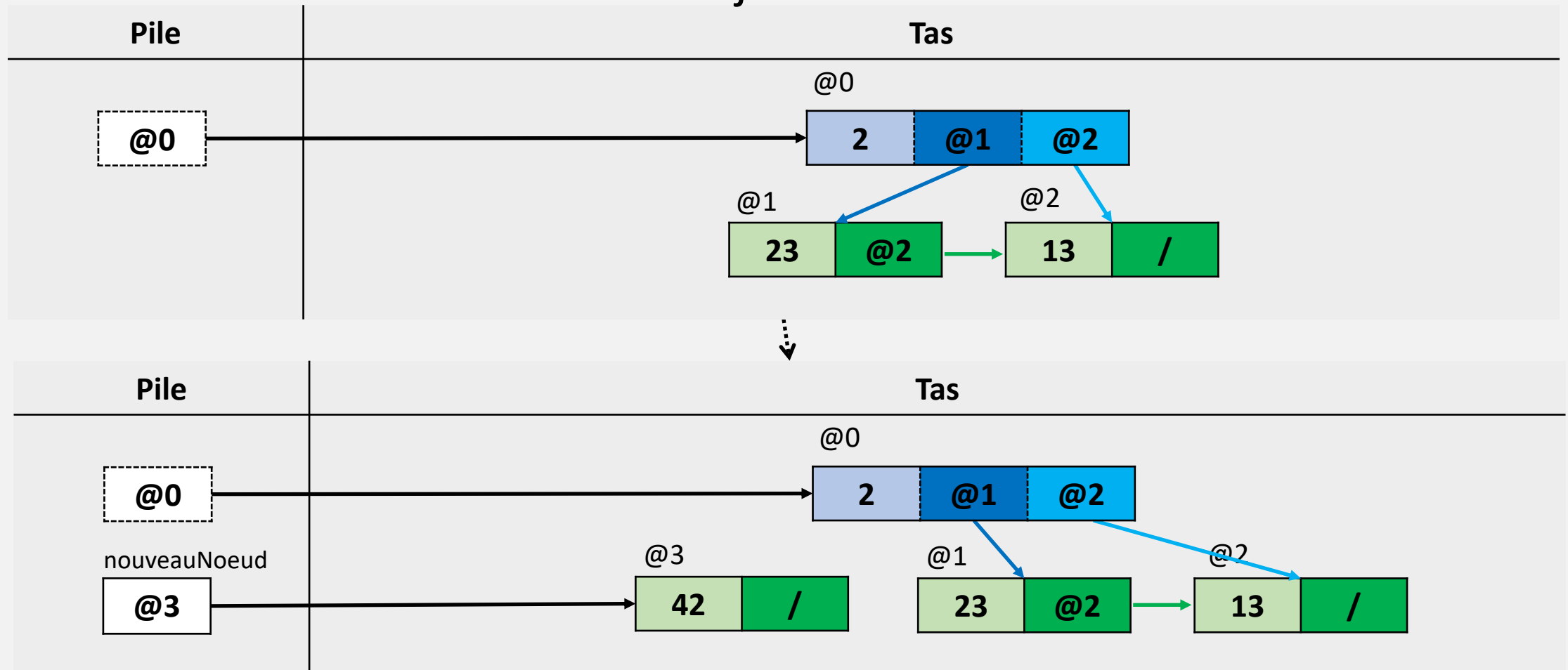


⇓ Ajouter 42 en début de liste



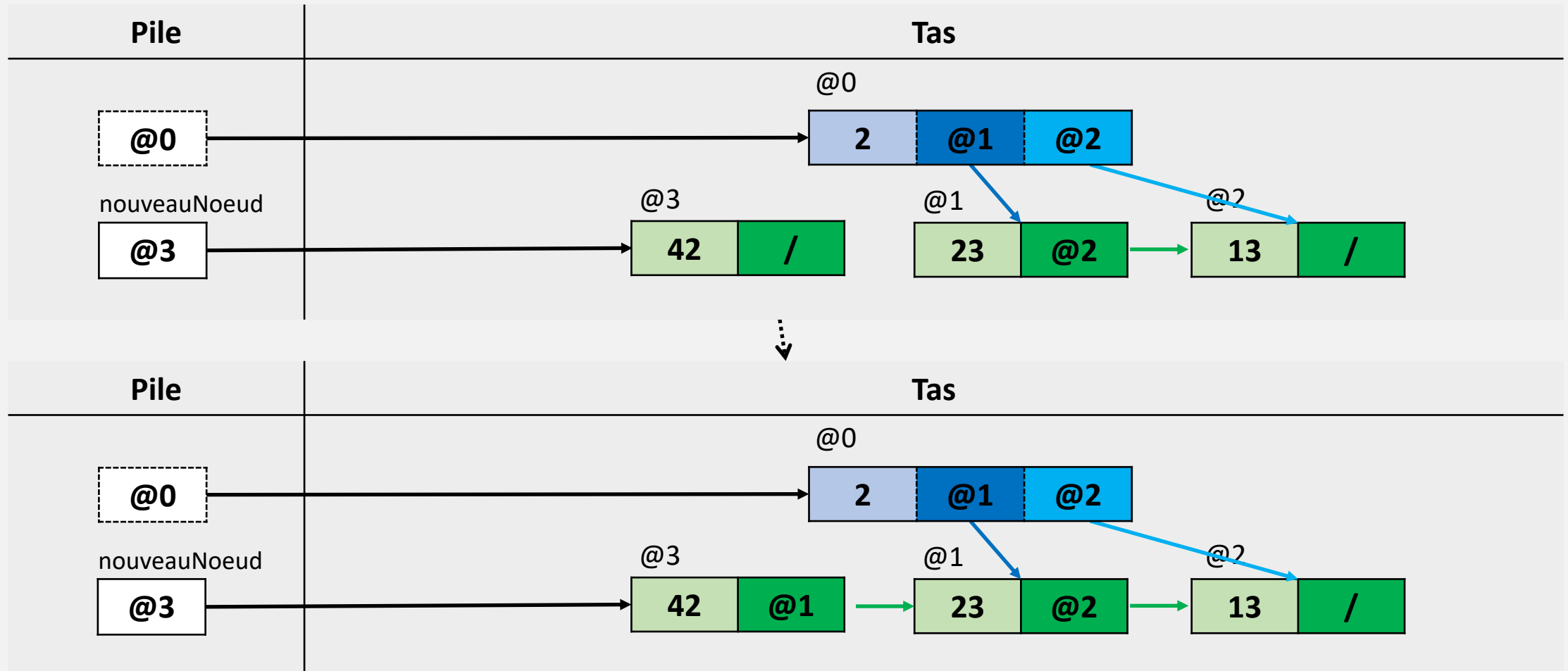
Opération – Ajout en début de liste

- Créer le nœud avec la valeur à ajouter



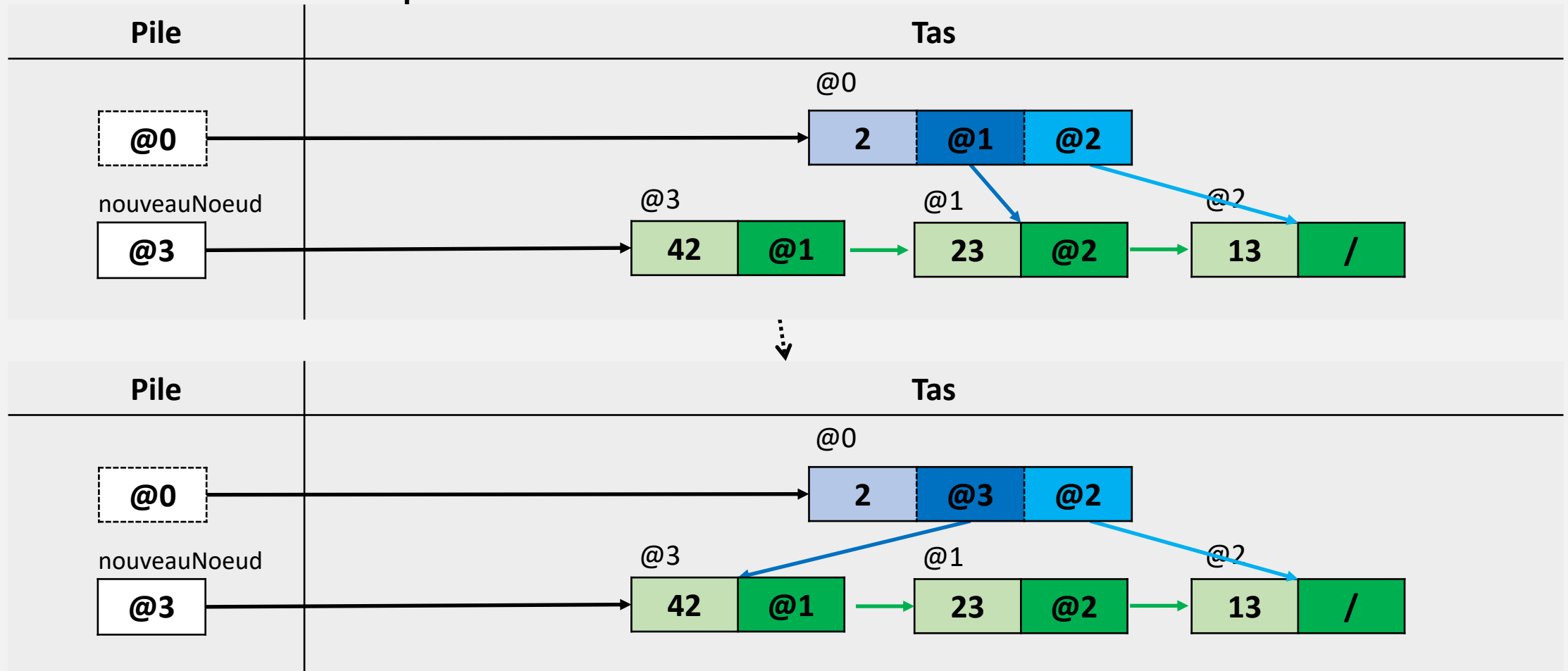
Opération – Ajout en début de liste

- Modification du suivant du nouveau nœud



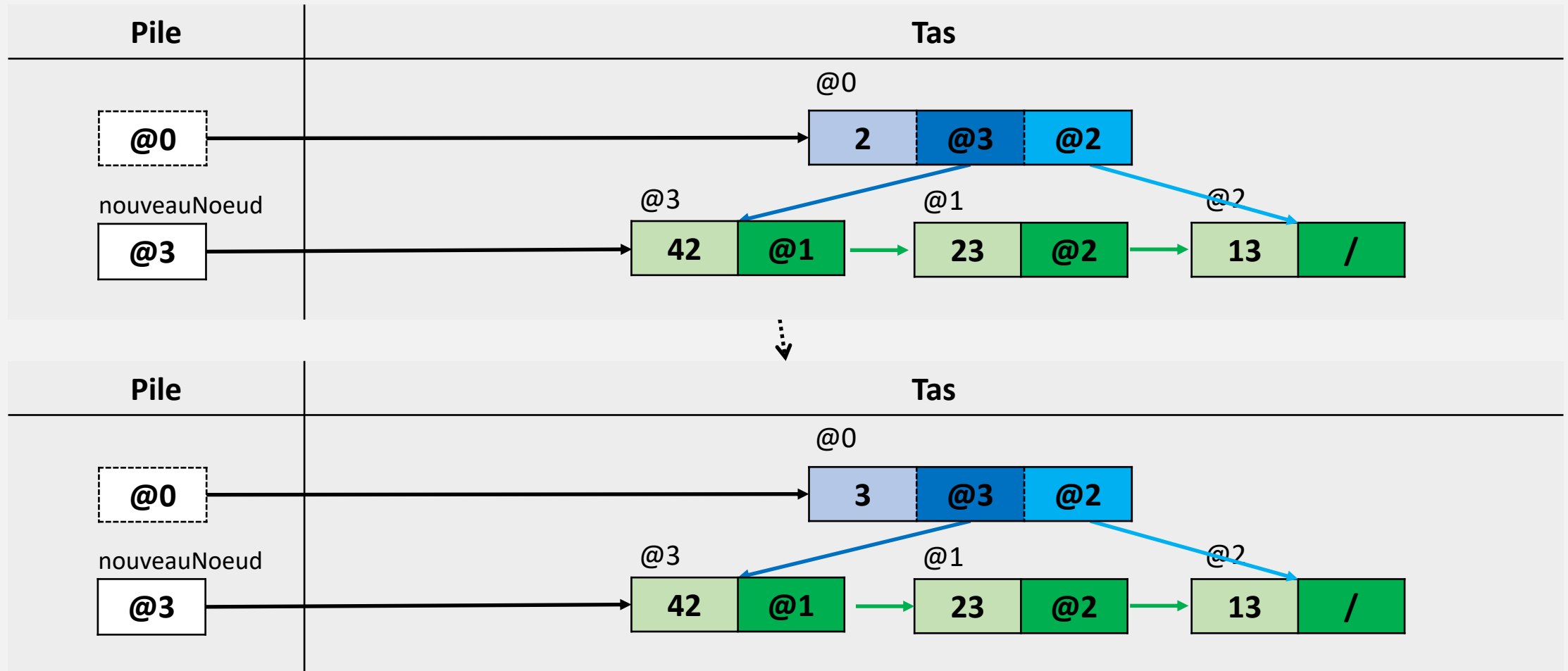
Opération – Ajout en début de liste

- Modification du premier nœud de la liste



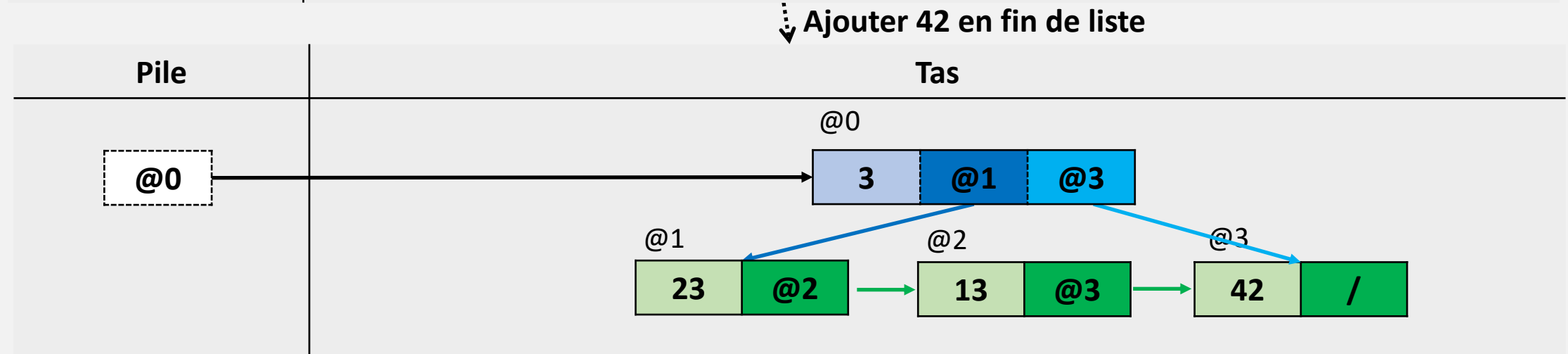
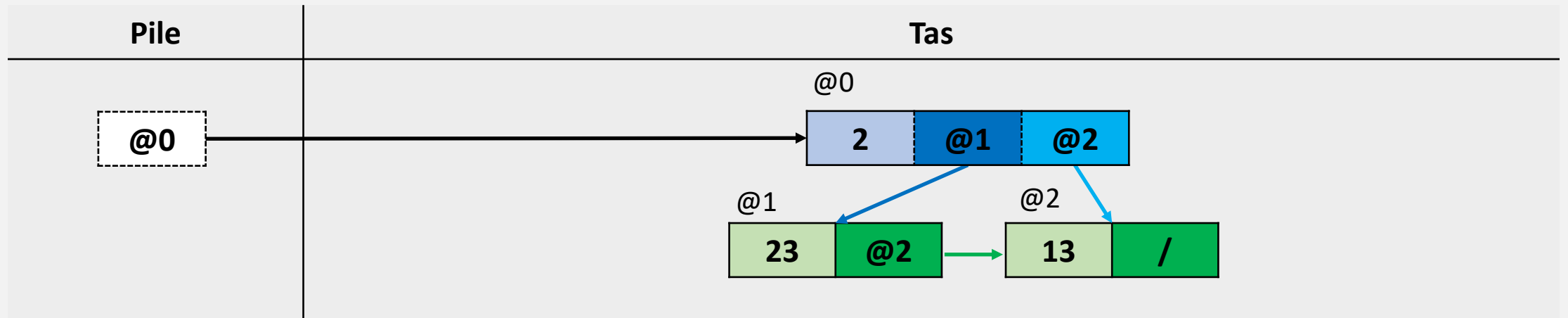
Opération – Ajout en début de liste

- Modification du nombre d'éléments de la liste



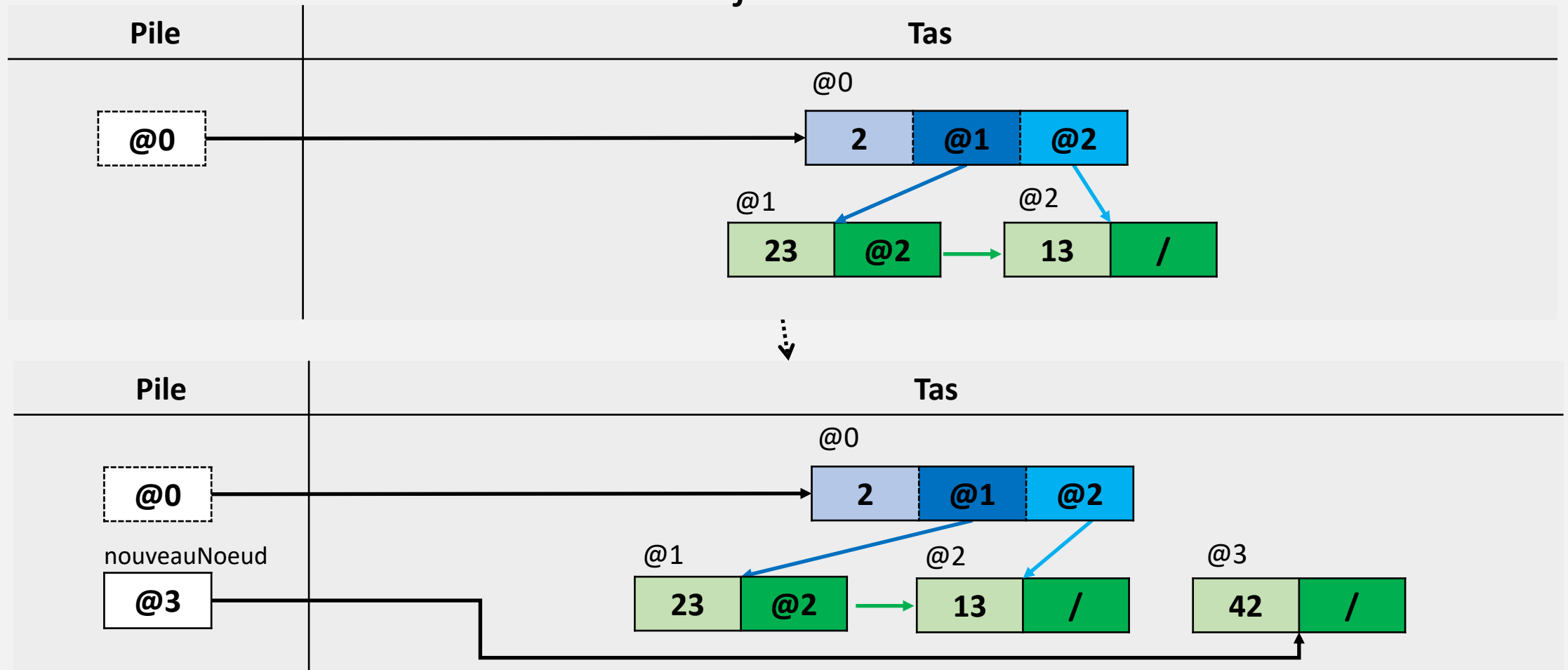
Opération – Ajout en fin de liste

- Idée



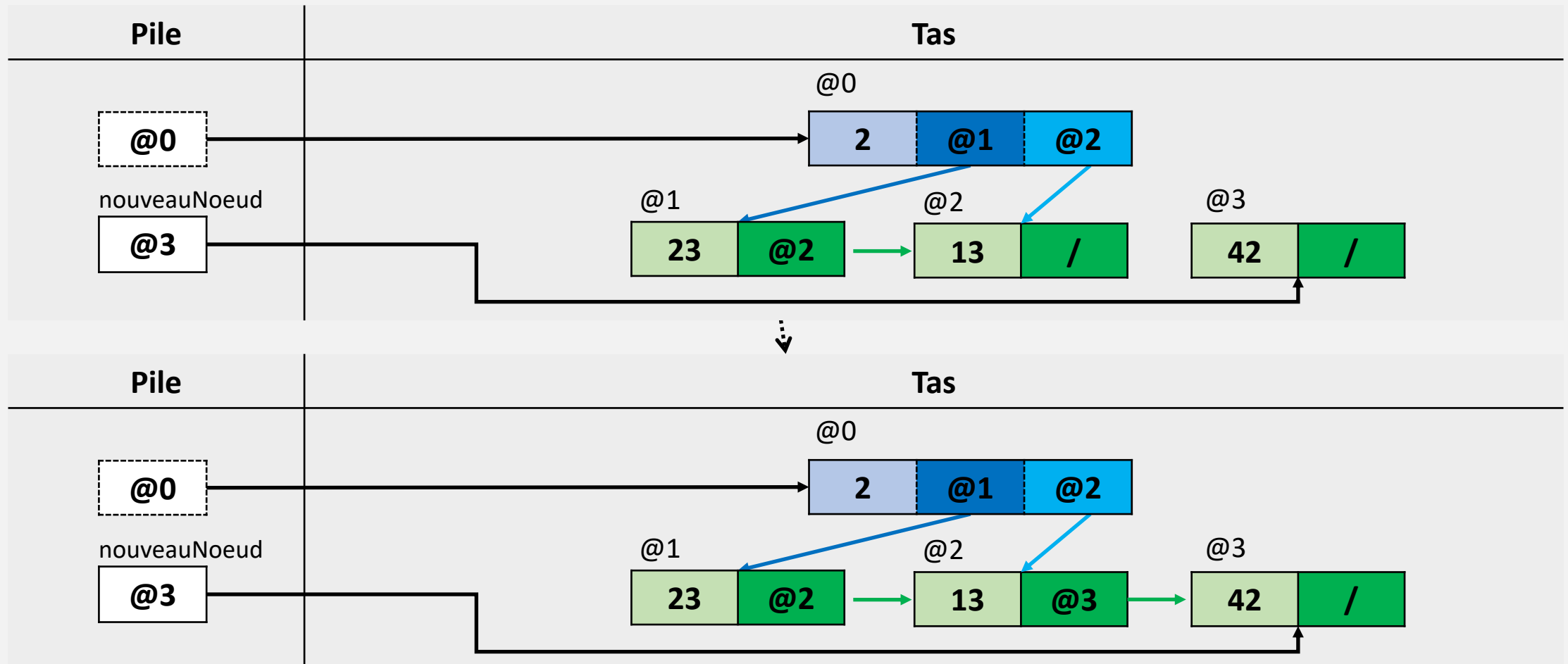
Opération – Ajout en fin de liste

- Créer le nœud avec la valeur à ajouter



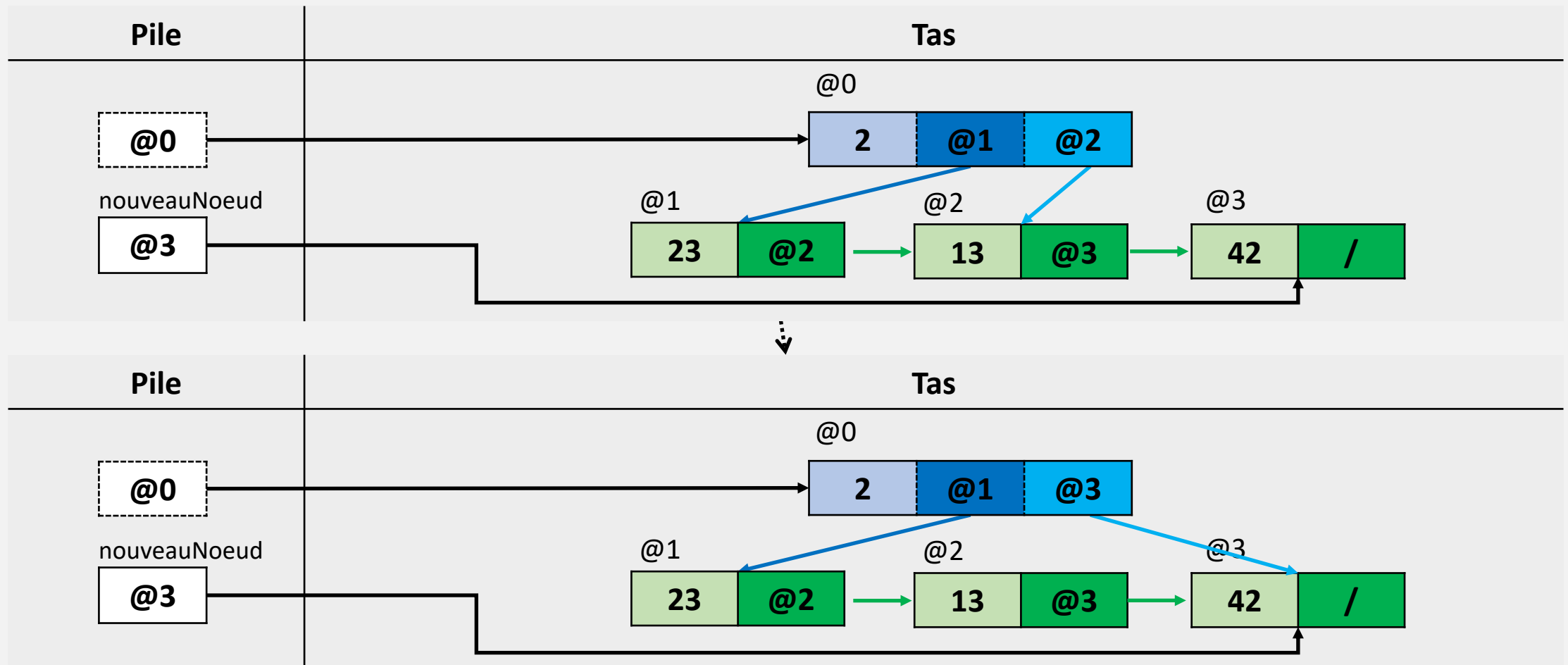
Opération – Ajout en fin de liste

- Modifier le suivant du dernier nœud pour le nouveau nœud : ici nous utilisons la référence du dernier nœud afin d'éviter un parcours



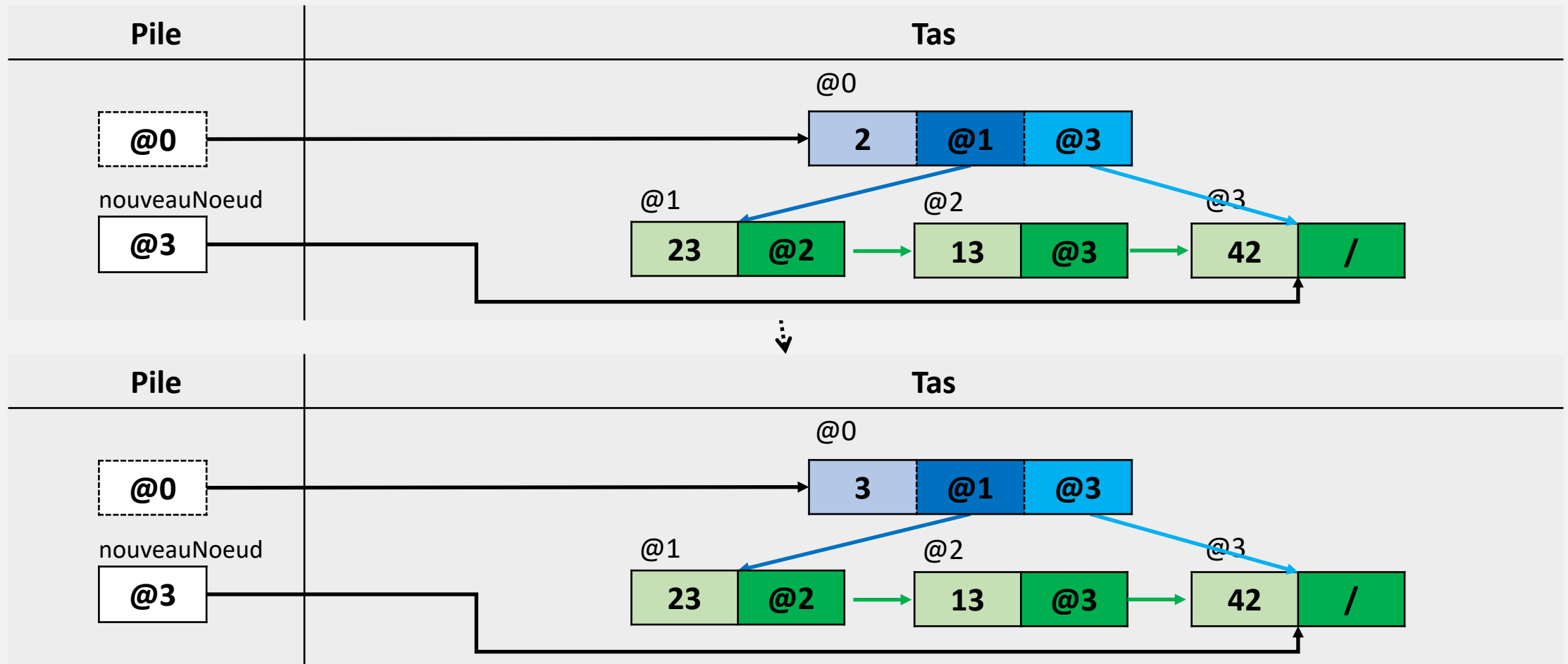
Opération – Ajout en fin de liste

- Modifier la référence vers le dernier nœud de la liste



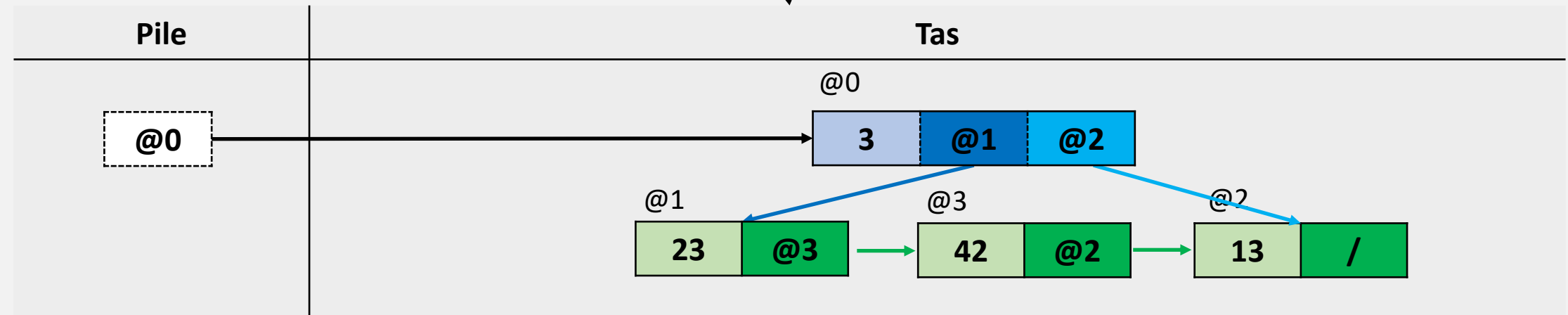
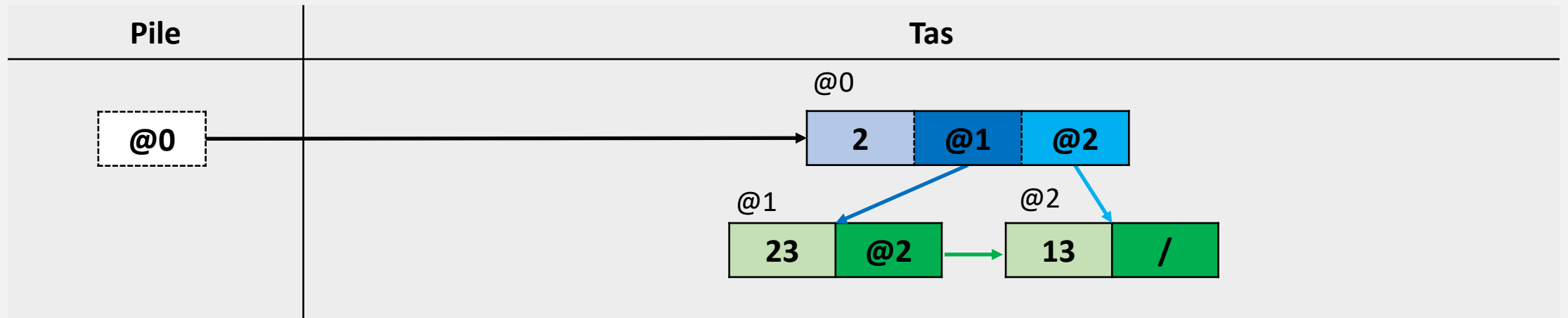
Opération – Ajout en fin de liste

- Modification du nombre d'éléments de la liste



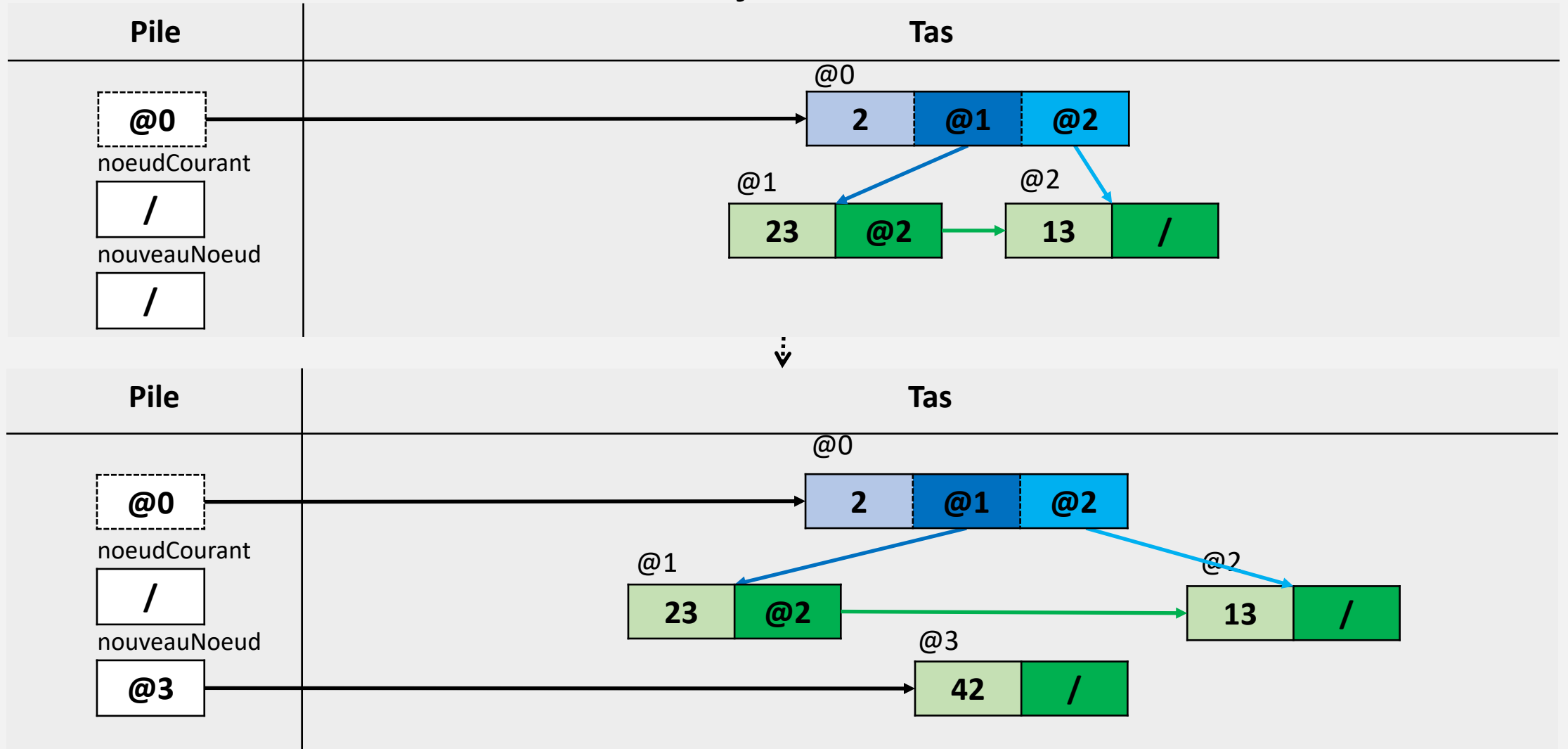
Opération – Insertion

- Idée



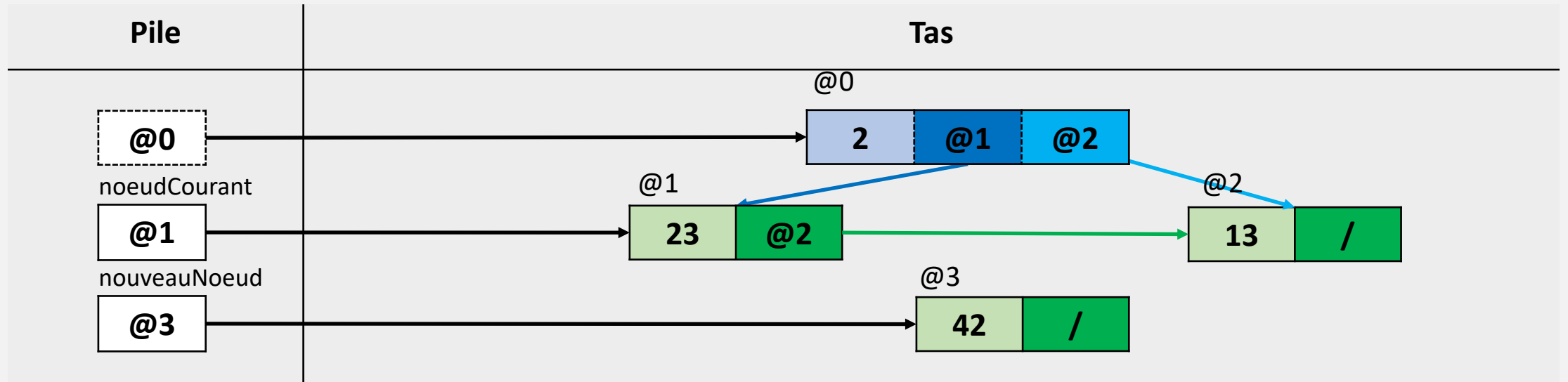
Opération – Insertion

- Créer le nœud avec la valeur à ajouter



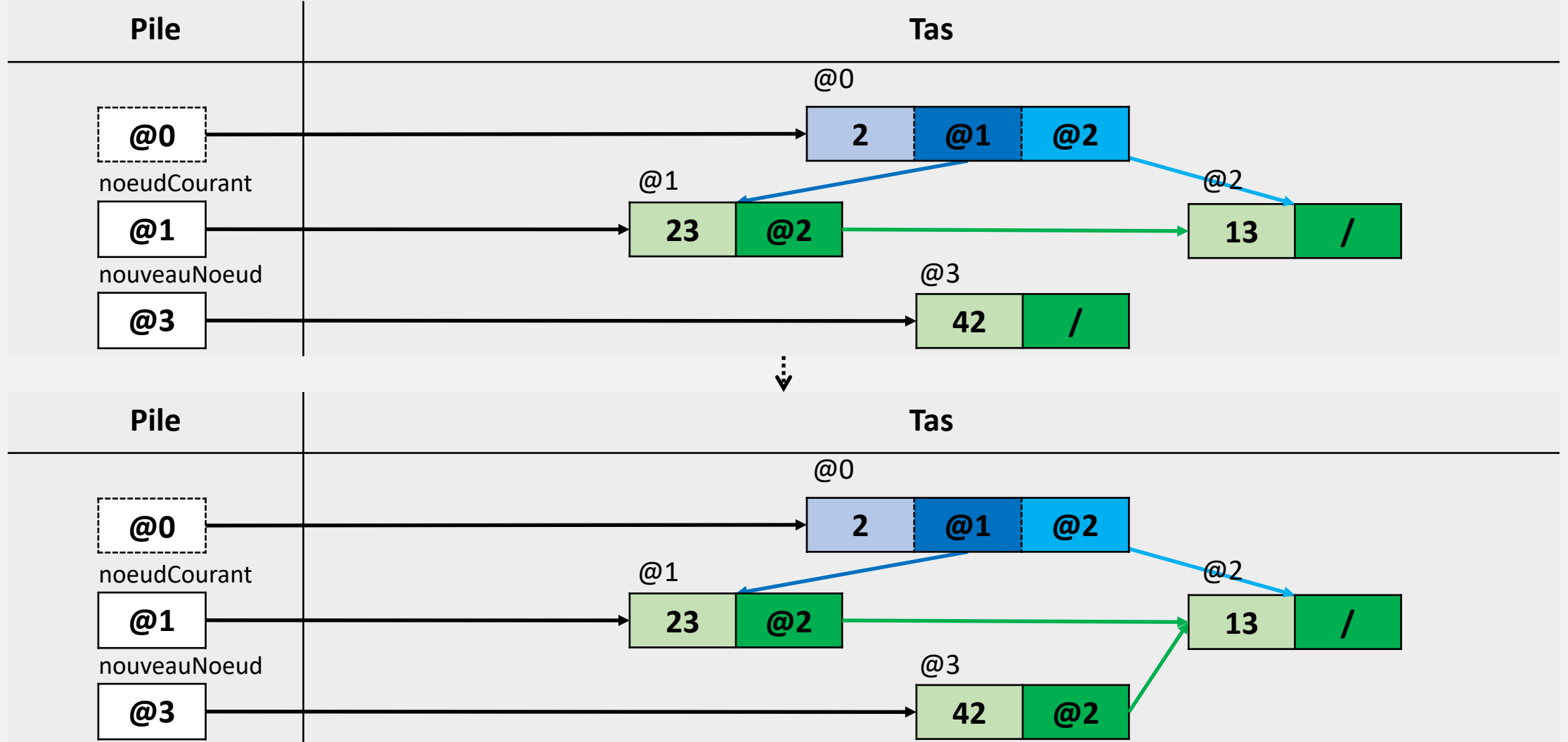
Opération – Insertion

- Recherche du nœud précédent l'insertion



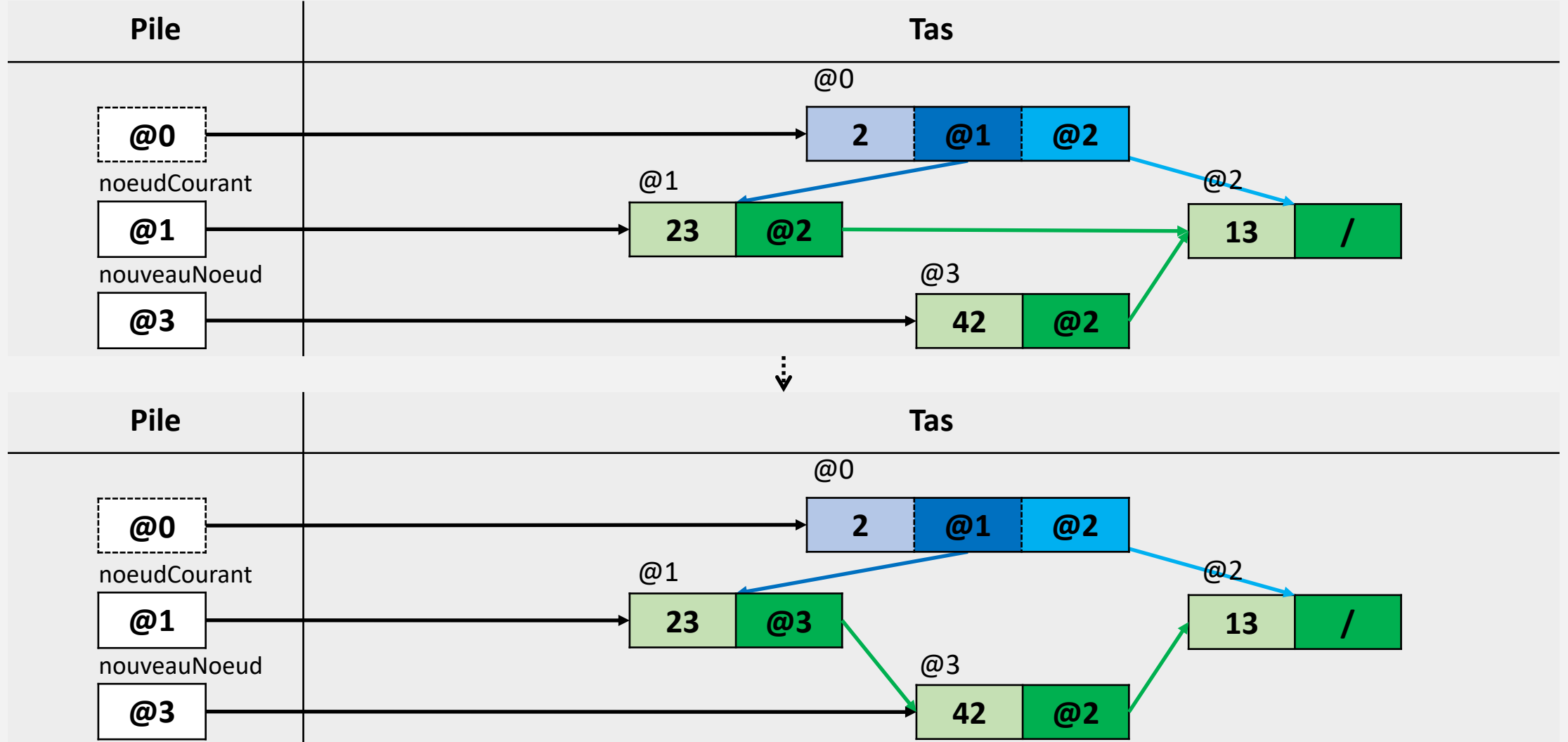
Opération – Insertion

- Modifier la référence du suivant du nouveau nœud



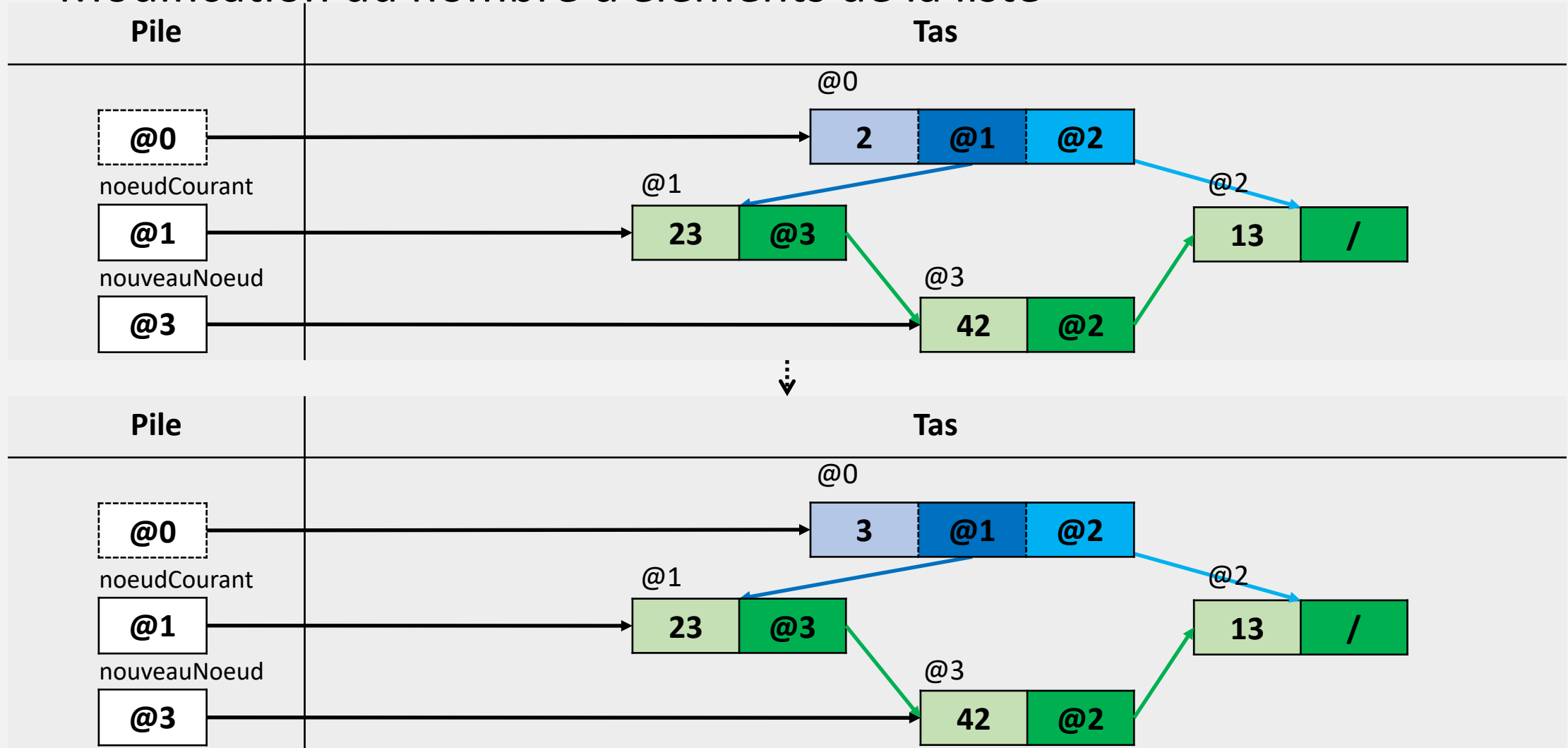
Opération – Insertion

- Modifier la référence du suivant du nouveau nœud



Opération – Insertion

- Modification du nombre d'éléments de la liste



Opération – Insertion – Cas particuliers

- L'insertion en position 0 doit être traitée comme un ajout en début
- L'insertion en position « NombreElements » doit être traitée comme un ajout en fin