



Université Sultan Moulay Slimane
Faculté Polydisciplinaire **Khouribga**



Sciences Mathématiques et Informatique

Structures de Données

Chapitre 3 : Listes

Pr. Ibtissam Bakkouri

i.bakkouri@usms.ma

Année Universitaire : **2022/2023**

Plan

- 1 Généralités sur les listes
- 2 Liste simplement chaînée
- 3 Liste doublement chaînée
- 4 Liste circulaire

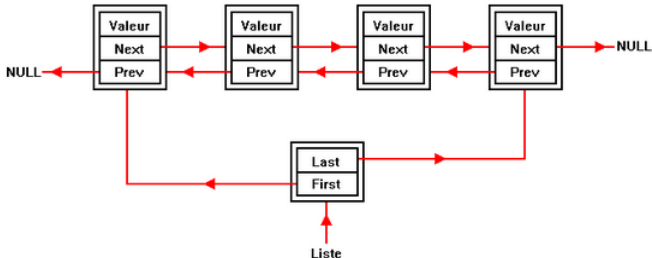
Généralités sur les listes

Les listes sont l'une des structures de données les plus fondamentales en informatique. Elles sont utilisées pour stocker et organiser des données de manière ordonnée et flexible, ce qui les rend utiles dans de nombreuses applications informatiques. Les listes sont constituées d'éléments qui peuvent être de différents types, tels que des nombres, des chaînes de caractères, des objets, etc.

Ces éléments peuvent être insérés, supprimés ou modifiés selon les besoins, offrant ainsi une grande flexibilité dans la manipulation des données.

Généralités sur les listes

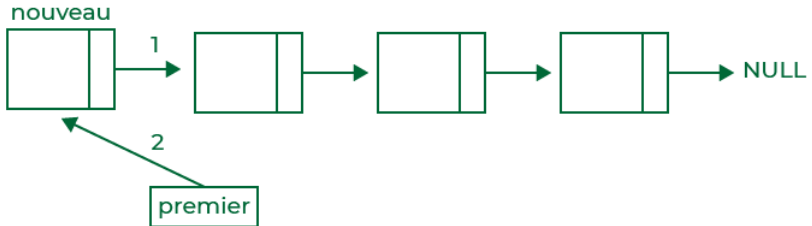
Les listes peuvent être implémentées de différentes manières, telles que des tableaux dynamiques ou des listes chaînées. Chacune de ces méthodes a ses avantages et ses inconvénients en termes de performances et d'utilisation de la mémoire, mais toutes deux offrent des fonctionnalités essentielles pour la manipulation de données.



Liste simplement chaînée

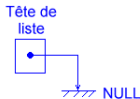
Liste simplement chaînée

Les listes chaînées sont des structures de données dynamiques qui permettent de stocker une collection d'éléments de manière flexible. Chaque élément, appelé nœud, contient une valeur et un pointeur vers le nœud suivant de la liste.

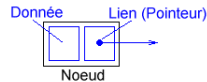


Liste simplement chaînée

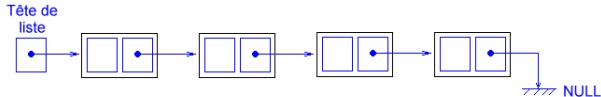
Liste vide :



Légende :



Exemple de liste non vide :



Dans cette liste, le premier nœud est appelé nœud tête, et il contient une valeur nulle pour indiquer le début de la liste. Le nœud suivant, appelé nœud 1, contient une valeur et un pointeur vers le nœud suivant, nœud 2. Enfin, le nœud 4 contient une valeur et un pointeur nul pour indiquer la fin de la liste.

Liste simplement chaînée

En C, un noeud sera représenté sous la forme d'une structure contenant une ou plusieurs informations et un pointeur (le lien) vers le noeud suivant dans la liste.

Pour déclarer une structure Noeud dont un des champs est un pointeur sur un Noeud on utilise ce code suivant:

```
typedef struct Noeud  
{  
int donnée;  
struct Noeud * suivant;  
} Noeud;
```

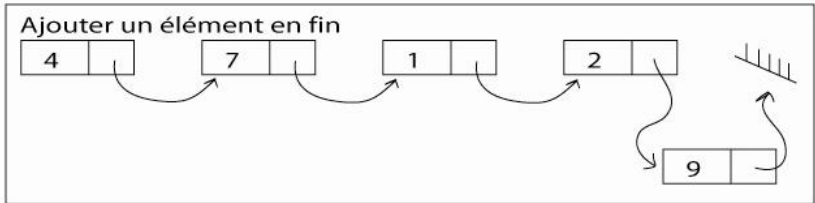
Liste simplement chaînée

Quelles sont les opérations de base sur les listes chaînées?

Les opérations de base sur les listes chaînées comprennent **l'ajout** de nœuds au début ou à la fin de la liste, **la suppression** de nœuds de la liste, **le parcours** de la liste et **la recherche** d'un nœud dans la liste. La navigation et la manipulation des listes chaînées se font à l'aide de pointeurs, ce qui permet une grande flexibilité dans la manipulation de la liste.

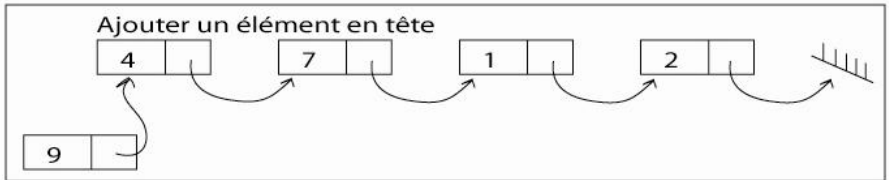
Liste simplement chaînée

Ajout d'un nœud à la fin de la liste : Pour ajouter un nœud à la fin de la liste, il faut créer un nouveau nœud et mettre à jour le pointeur du dernier nœud de la liste pour qu'il pointe vers le nouveau nœud.



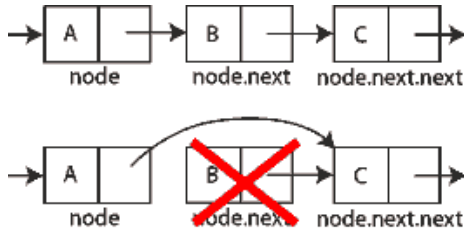
Liste simplement chaînée

Ajout d'un nœud au début de la liste : Pour ajouter un nœud au début de la liste, il faut créer un nouveau nœud et mettre à jour le pointeur de ce nouveau nœud pour qu'il pointe vers l'ancien nœud tête de la liste, puis mettre à jour le pointeur de la tête de liste pour qu'il pointe vers le nouveau nœud.



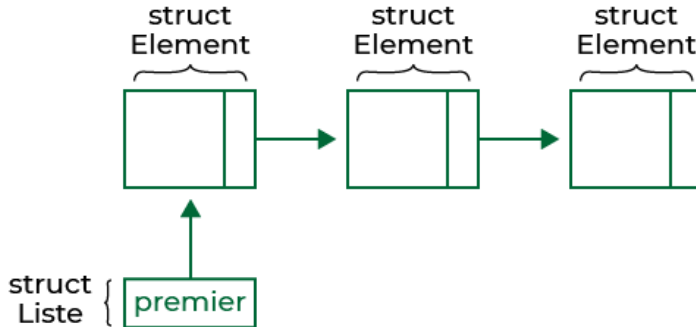
Liste simplement chaînée

Suppression d'un nœud de la liste : Pour supprimer un nœud de la liste, il faut trouver le nœud précédent le nœud à supprimer, mettre à jour son pointeur pour qu'il pointe vers le nœud suivant, puis libérer la mémoire utilisée par le nœud à supprimer.



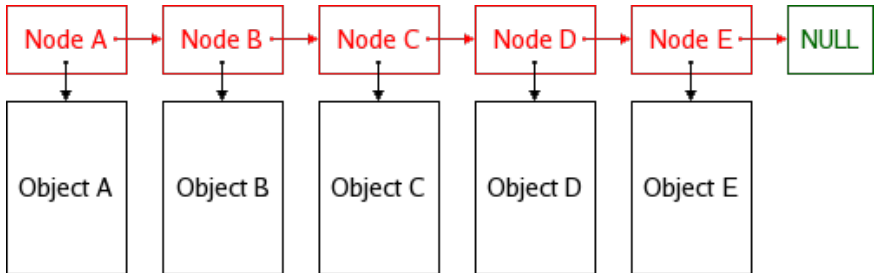
Liste simplement chaînée

Parcours de la liste : Pour parcourir la liste, il faut commencer par la tête de liste et suivre les pointeurs jusqu'à atteindre le dernier nœud de la liste.



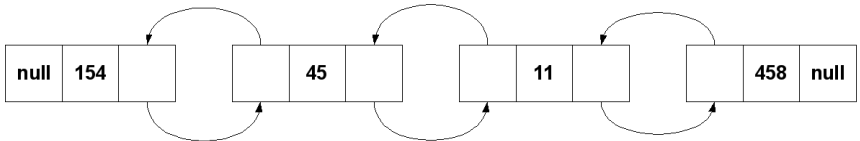
Liste simplement chaînée

Recherche d'un nœud dans la liste : Pour rechercher un nœud dans la liste, il faut parcourir la liste en comparant la valeur de chaque nœud avec la valeur recherchée jusqu'à trouver le nœud correspondant.



Liste doublement chaînée

Une liste doublement chaînée est une structure de données qui permet de stocker une collection d'éléments de manière ordonnée. Chaque élément de la liste est lié à son prédécesseur et à son successeur, ce qui permet de parcourir la liste dans les deux sens.

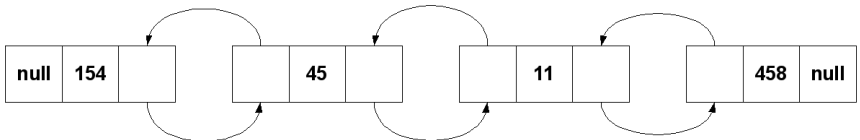


Élément de tête

Élément de queue

Liste doublement chaînée

La liste doublement chaînée est souvent utilisée dans les algorithmes de tri et dans la gestion de mémoire dynamique. Elle est également utile lorsque l'on a besoin d'accéder fréquemment à des éléments de la liste à la fois en avant et en arrière.

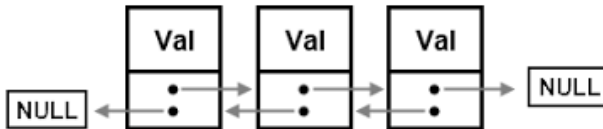


Elément de tête

Elément de queue

Liste doublement chaînée

La structure d'une liste doublement chaînée est constituée de nœuds, chacun ayant deux pointeurs: un pointeur vers l'élément précédent et un pointeur vers l'élément suivant. Le premier élément de la liste est lié au nœud NULL par son pointeur précédent, et le dernier élément est lié au nœud NULL par son pointeur suivant.



Liste doublement chaînée

Voici un exemple de structure de nœud pour une liste doublement chaînée:

```
struct node  
{  
  int data;  
  struct node* prev;  
  struct node* next;  
};
```

Liste doublement chaînée

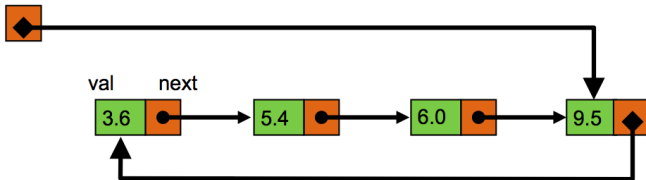
Les opérations de base sur les listes doublement chaînées comprennent :

- **Insertion** : l'ajout d'un nouvel élément à la liste. Il peut être inséré au début, à la fin ou à un emplacement donné.
- **Suppression** : la suppression d'un élément existant de la liste. L'élément peut être supprimé de n'importe où dans la liste.
- **Parcourir** : accéder à chaque élément de la liste dans l'ordre, du début à la fin ou vice versa.
- **Recherche** : la recherche d'un élément spécifique dans la liste.
- **Accéder à un élément spécifique** : l'accès direct à un élément de la liste en utilisant un index ou une clé.
- **Modification** : modifier la valeur d'un élément existant dans la liste.

Liste circulaire

Une liste circulaire est une structure de données qui contient une séquence d'éléments où chaque élément pointe vers l'élément suivant et précédent. Lorsqu'on arrive au dernier élément de la liste, l'élément suivant pointe vers le premier élément, créant ainsi une boucle.

*La liste circulaire est manipulée
par un simple pointeur sur la queue*

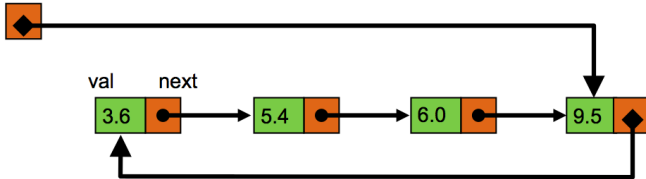


*Circularité : le maillon de queue pointe
sur le maillon de tête*

Liste circulaire

Les listes circulaires sont souvent utilisées dans les programmes où une itération continue sur les éléments est nécessaire, comme les jeux vidéo ou les lecteurs audio.

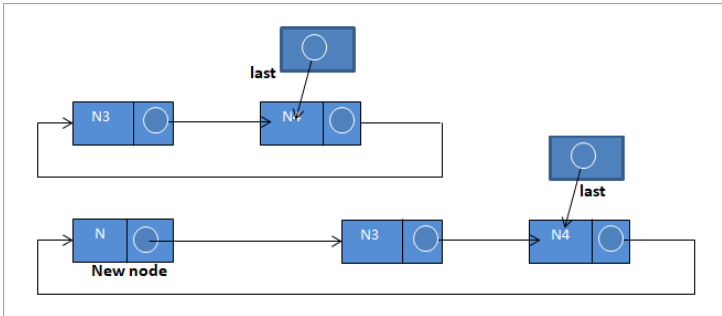
*La liste circulaire est manipulée
par un simple pointeur sur la queue*



*Circularité : le maillon de queue pointe
sur le maillon de tête*

Liste circulaire

Les opérations de base sur les listes circulaires chaînées sont similaires à celles des listes chaînées simples, mais avec quelques différences dues à la nature circulaire de la structure.



Liste circulaire

Voici les opérations de base sur les listes circulaires chaînées :

- Création d'une liste circulaire chaînée : Il faut créer un premier nœud et le lier à lui-même pour former la boucle circulaire. Ensuite, d'autres nœuds peuvent être créés et insérés dans la liste en fonction des besoins.
- Insertion d'un élément : L'insertion d'un élément dans une liste circulaire chaînée peut se faire de deux manières : en tête ou en queue de liste. Si l'insertion se fait en tête, le nouveau nœud devient la nouvelle tête de liste. Si l'insertion se fait en queue, le nouveau nœud devient la nouvelle queue de liste.

Liste circulaire

- Suppression d'un élément : La suppression d'un élément dans une liste circulaire chaînée peut se faire également de deux manières : en tête ou en queue de liste. Si la suppression se fait en tête, le nœud suivant devient la nouvelle tête de liste. Si la suppression se fait en queue, le nœud précédent devient la nouvelle queue de liste.
- Parcours d'une liste circulaire chaînée : Le parcours d'une liste circulaire chaînée se fait également en boucle, en partant de la tête de liste et en se déplaçant de nœud en nœud jusqu'à revenir à la tête. Il est important de prendre en compte la condition d'arrêt de la boucle pour éviter une boucle infinie.

Liste circulaire

- Recherche d'un élément : La recherche d'un élément dans une liste circulaire chaînée se fait de la même manière que dans une liste chaînée simple, en parcourant la liste jusqu'à trouver l'élément recherché ou atteindre la fin de la liste sans le trouver.

En conclusion, les listes sont une structure de données fondamentale en informatique. Elles sont utilisées pour stocker et organiser des données de manière ordonnée et flexible. Les listes sont constituées d'éléments qui peuvent être de différents types et qui peuvent être insérés, supprimés ou modifiés.