

Algorithmique et structures de données

Loïc Demange

`loic.demange@etud.univ-paris8.fr`

26 novembre 2021



Liste chaînée

La liste chaînée est une structure d'éléments ordonnés où chaque élément pointe sur l'élément d'après.

Contrairement aux tableaux, la taille n'est pas fixe et les éléments ne sont pas forcément contiguës en mémoire.

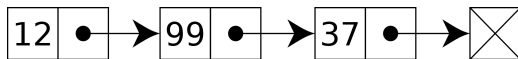


Figure: Liste chaînée (wikipédia)

Remarque Il existe aussi la liste doublement chaînée, où chaque élément pointe sur l'élément d'avant et d'après.

Chaque élément d'une liste chaînée est appelé noeud.

- Comment est défini un noeud ?

Chaque élément d'une liste chaînée est appelé noeud.

- Comment est défini un noeud ?

Un noeud est défini par une valeur, et un lien vers le noeud suivant.

On peut donc définir un noeud de cette façon en C, en déclarant une structure.

```
typedef struct noeud {  
    int valeur;  
    struct noeud * suivant;  
} noeud_t;
```

- Comment stocke-t-on une liste chaînée ?

- Comment stocke-t-on une liste chaînée ?

On garde le premier élément de la chaîne, et on peut itérativement la parcourir avec le lien vers le noeud suivant.

- Comment insere-t-on un élément dans une liste chaînée ?

- Comment insère-t-on un élément dans une liste chaînée ?

Si on insère un élément au début, il suffit de faire pointer le lien du noeud suivant du nouveau noeud vers le premier noeud actuel.

Si on insère un élément à la fin, il suffit de faire pointer le lien du noeud suivant du dernier noeud actuel vers le nouveau noeud.

Si on insère un élément au milieu, il suffit de faire pointer le lien du noeud suivant du noeud précédent vers le nouveau noeud, et faire pointer le lien du noeud suivant du noeud actuel vers le noeud suivant actuel.

Liste chaînée

- Comment insère-t-on un élément dans une liste chaînée ?

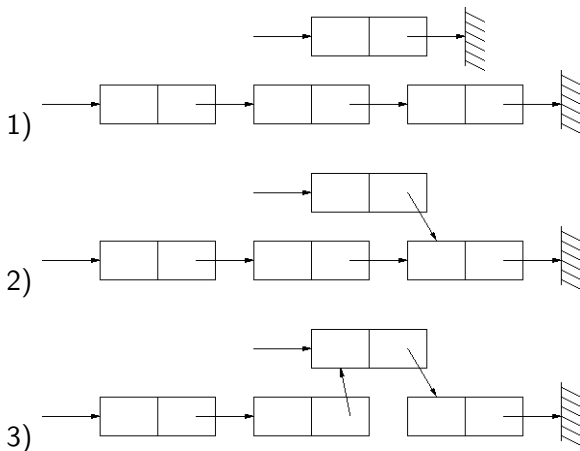


Figure: Insertion dans une liste chaînée (wikipédia)

- Comment supprime-t-on un élément dans une liste chaînée ?

- Comment supprime-t-on un élément dans une liste chaînée ?

Si on supprime le premier élément, il suffit de prendre l'élément suivant comme noeud principal.

Si on supprime le dernier élément, il suffit de faire pointer le lien de l'avant dernier noeud actuel vers aucun noeud.

Si on supprime un élément quelconque, il suffit de faire pointer le lien du noeud précédent du noeud qu'on souhaite supprimer vers le noeud suivant du noeud qu'on souhaite supprimer.

Remarque En C, ne pas oublier de désallouer les noeuds qu'on souhaite supprimer s'ils ont été alloués dynamiquement.

- Quelle est la complexité temporelle pour accéder à un noeud particulier d'une liste chaînée ?

- Quelle est la complexité temporelle pour accéder à un noeud particulier d'une liste chaînée ?

Pour accéder à un noeud en particulier, il faut nécessairement parcourir tous ses noeuds précédents.

Dans le pire des cas (dernier élément), on va parcourir toute la liste chaînée.

Donc si la taille de la liste chaînée (le nombre d'éléments de la liste) est n , on est en $\mathcal{O}(n)$.

Questions de compréhension :

- Quels sont les avantages et inconvénients d'une liste chaînée par rapport aux tableaux ?
- Quelle est la complexité temporelle de l'ajout d'élément à une liste ? De la suppression ?
- Une liste doublement chaînée permet-elle d'améliorer certaines manipulations par rapport à une liste chaînée classique ?

Objectif du TP : implémenter les listes chaînées avec les fonctions suivantes :

- Création et ajout de noeuds.
- Suppression de noeuds.
- Affichage de la liste.