Algorithmique et structures de données

Loïc Demange
loic.demange@etud.univ-paris8.fr

04-11-18 décembre 2020

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.

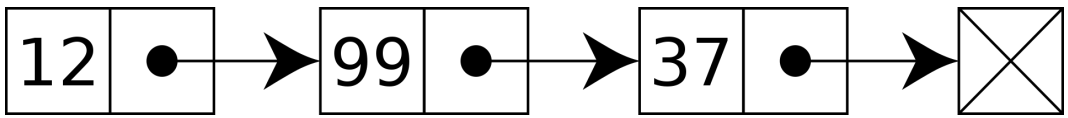

Figure: Liste chaînée (wikipédia)

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

Liste chaînée

Chaque élément d'une liste chaînée est appelé noeud.

- Comment est défini un noeud ?

Liste chaînée

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.

Liste chaînée

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;
```

Liste chaînée

- Comment stocke-t-on une liste chaînée ?

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.

Liste chaînée

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

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 ?

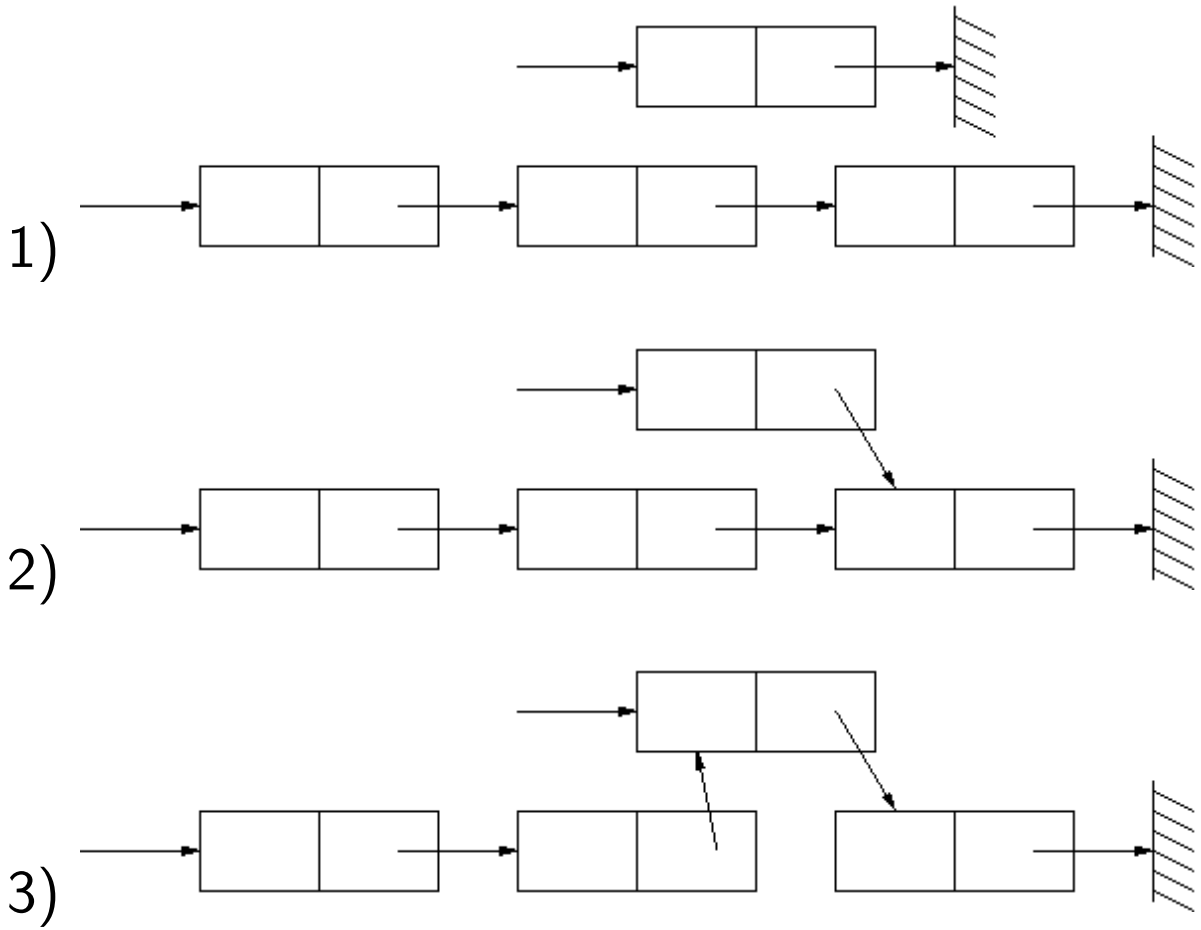

Figure: Insertion dans une liste chaînée (wikipédia)

Liste chaînée

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

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.

Liste chaînée

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

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 $\mathcal{O}(n)$.

Liste chaînée

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 ?

Piles et files

Les piles et les files sont des structures de données où chaque élément pointe sur l'élément d'après, et où on ne peut ajouter et retirer des éléments qu'à un certain endroit et à un certain ordre.

Dans le cas de la pile, on peut ajouter des éléments qu'en queue de structure, et le retrait se fera à partir de la queue (dernier arrivé premier parti, LIFO en anglais).

On peut comparer ça à une pile d'assiette : on récupère la première assiette, qui est celle qui a été posée en dernier.

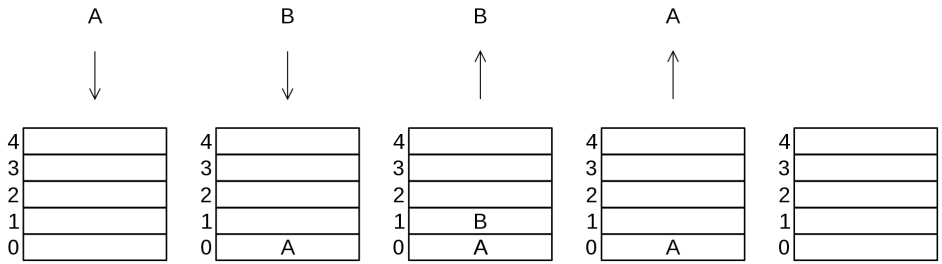

Figure: Pile (wikipédia)

Piles et files

Dans le cas de la file, on peut ajouter des éléments qu'en queue de structure, et le retrait se fera à partir de la tête (premier arrivé premier parti, FIFO en anglais).

On peut comparer ça à une queue dans un magasin : le premier arrivé sera le premier à passer à la caisse.

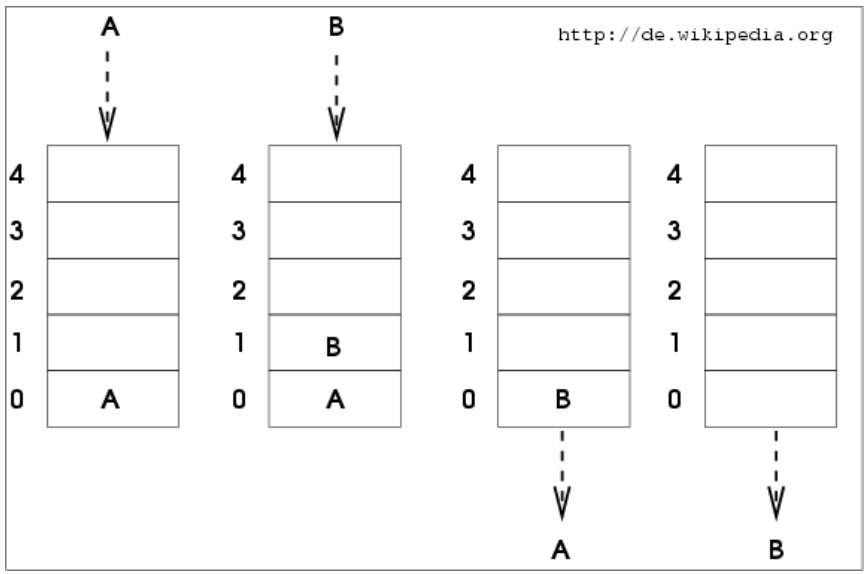

Figure: File (wikipédia)

Piles et files

- Comment implémenter une pile ?

Piles et files

- Comment implémenter une pile ?

On peut utiliser le même fonctionnement de la liste chaînée, en ne gardant que les méthodes pour ajouter un élément en fin de liste ainsi que celle pour retirer le dernier élément.

Piles et files

- Comment implémenter une file ?

Piles et files

- Comment implémenter une file ?

On peut utiliser le même fonctionnement de la liste chaînée, en ne gardant que les méthodes pour ajouter un élément en fin de liste ainsi que celle pour retirer le premier élément.

Piles et files

Question pour être à l'aise :

- Prendre le code C de la liste chaînée et faire l'implémentation de la pile et de la file.

Arbres

Les arbres sont une structure de données sous forme d'un graphe orienté sans cycle possédant une unique racine.
Chaque élément est appelé noeud, ses successeurs sont appelés des fils, et on appelle feuille un élément n'ayant pas de fils.

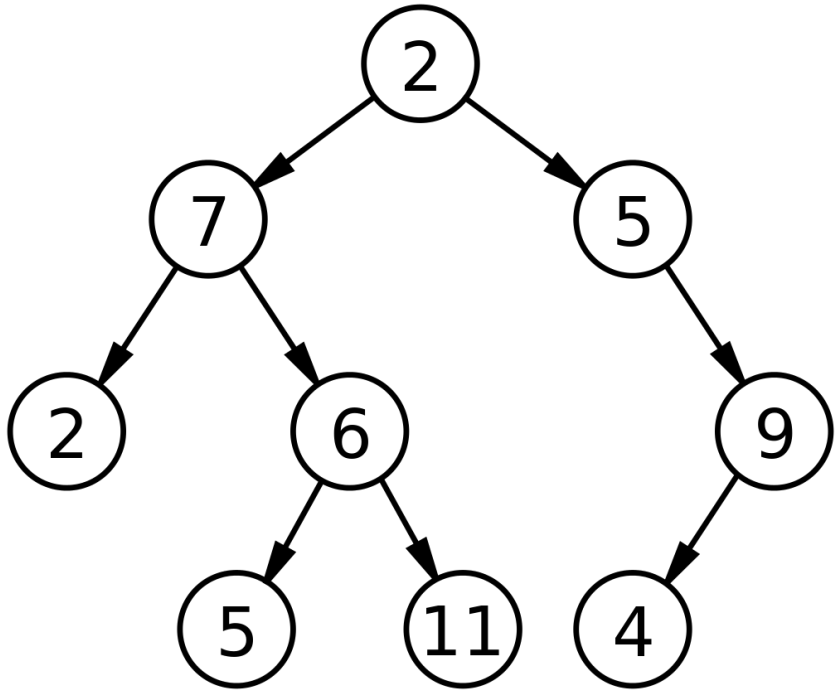

Figure: Arbre binaire (wikipédia)

Arbres

Il existe différents types d'arbres, mais nous allons nous concentrer sur l'arbre binaire, où chaque noeud ne peut avoir que deux fils maximum.

- Comment est défini un noeud dans un arbre binaire ?

Arbres

Il existe différents types d'arbres, mais nous allons nous concentrer sur l'arbre binaire, où chaque noeud ne peut avoir que deux fils maximum.

- Comment est défini un noeud dans un arbre binaire ?

Un noeud est défini par une valeur, un lien vers le successeur droite et un lien vers le successeur gauche.

Arbres

- Comment stocke-t-on un arbre ?

Arbres

- Comment stocke-t-on un arbre ?

On garde le premier élément de l'arbre (la racine), et on peut itérativement la parcourir l'arbre avec les liens vers les successeurs.

Arbres

De la même façon que pour les listes chaînées, on peut définir une struture en C pour les noeuds.

```
typedef struct noeud {
    int valeur;
    struct noeud * succ_droit;
    struct noeud * succ_gauche;
} noeud_t;
```

Arbres

L'arbre binaire de recherche (ou ABR) est un arbre binaire où les valeurs sont agencées de façon à être efficace dans la recherche de valeurs. Conceptuellement, un noeud a pour successeur gauche une valeur inférieure à la sienne et pour successeur droit une valeur supérieure à la sienne.

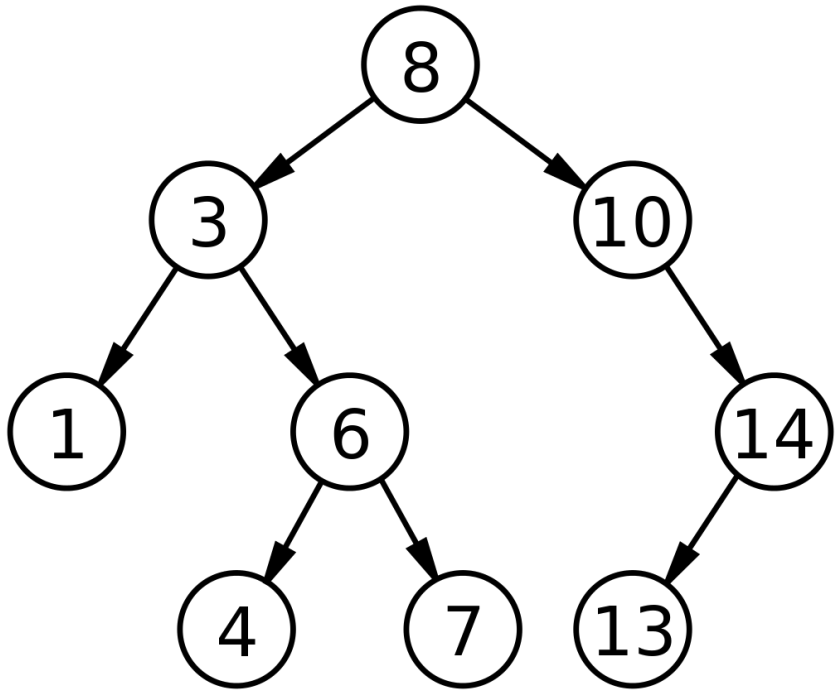

Figure: Arbre binaire de recherche (wikipédia)

Arbres

- Comment cherche-t-on un élément dans un arbre binaire de recherche ?

Arbres

- Comment cherche-t-on un élément dans un arbre binaire de recherche ?

On peut utiliser la dichotomie : si l'élément qu'on cherche est plus grand que la valeur du noeud, on part du côté du successeur droite ; sinon on part du côté du successeur gauche.

Arbres

- Quelle est la complexité temporelle de la recherche dans un arbre binaire de recherche ?

Arbres

- Quelle est la complexité temporelle de la recherche dans un arbre binaire de recherche ?

Si l'arbre est équilibré, on est en moyenne en $\mathcal{O}(\log_2(n))$, n étant le nombre de noeuds de l'arbre.
Dans le cas contraire, l'arbre ressemble à une liste chaînée, le pire cas serait donc en $\mathcal{O}(n)$ (on parcourt tous les noeuds pour trouver la valeur).

Remarque Ici on ne va pas parler de la manière d'équilibrer un arbre. Il existe toutefois divers algorithmes pour cela, dont certains à base de rotation.

Arbres

- Comment insere-t-on un élément dans un arbre binaire de recherche?

Arbres

- Comment insere-t-on un élément dans un arbre binaire de recherche?

On recherche une feuille de l'arbre. Une fois trouvée, si la valeur du noeud est inférieure à la valeur de la feuille, on le met en successeur gauche ; sinon on le met en successeur droit.

Arbres

- Quelle est la complexité temporelle de l'insertion dans un arbre binaire de recherche ?

Arbres

- Quelle est la complexité temporelle de l'insertion dans un arbre binaire de recherche ?

De la même façon que la recherche, la complexité est en moyenne en $\mathcal{O}(\log_2(n))$, sauf si l'arbre est déséquilibré. Auquel cas la complexité est au pire en $\mathcal{O}(n)$.

Arbres

- Comment supprime-t-on un élément dans un arbre binaire de recherche ?

Arbres

- Comment supprime-t-on un élément dans un arbre binaire de recherche ?

Si on supprime une feuille, il suffit de faire pointer le lien du prédécesseur vers aucun noeud.
Si on supprime un noeud avec un seul fils, il suffit de faire pointer le prédécesseur vers le fils du noeud actuel.
Si on supprime un noeud avec deux fils, on échange la valeur du noeud actuel avec la valeur la plus importante de la sous-arborescence gauche. Une fois fait, on n'a plus qu'à supprimer le noeud à sa nouvelle position, où il ne sera qu'une feuille ou un noeud avec un seul fils gauche.

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

Arbres

- Quelle est la complexité temporelle de la suppression dans un arbre binaire de recherche ?

- Quelle est la complexité temporelle de la suppression dans un arbre binaire de recherche ?

De la même façon que la recherche, la complexité est en moyenne en $\mathcal{O}(\log_2(n))$, sauf si l'arbre est déséquilibré. Auquel cas la complexité est au pire en $\mathcal{O}(n)$.

Remarque Utiliser la fonction de suppression va potentiellement déséquilibrer l'arbre.