

Algorithmique avancée

Listes chaînées

Frédéric Guyomarch

Université de Lille1
IUT-A de Lille

2020/2021 - Semestre 3

Introduction

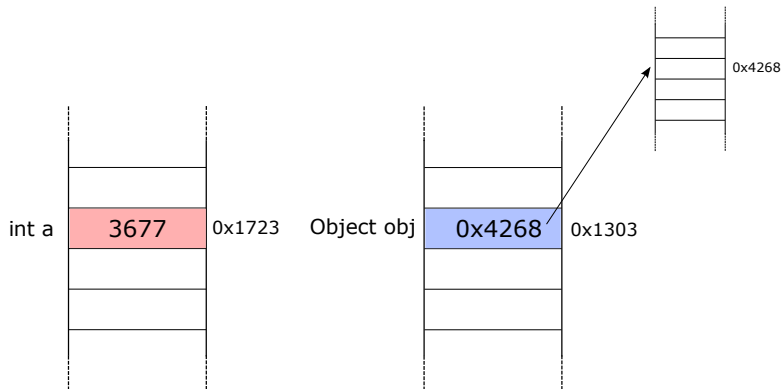
Tableaux :

- Taille fixe (statique)
 - Espace alloué parfois trop grand
 - Ou trop petit
- Accès direct
- Solution : Liste ? (dynamique)

TDA Liste

- Type de données abstrait Liste : *Suite ordonnée de noms, de mots, de chiffres, de nombres, de symboles, etc.*
- Structures de données qui l'implémente : *Les listes chaînées, les tableaux, etc.*

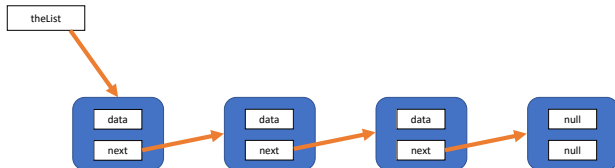
Références et valeurs



Liste chaînée

- Accès itératif
- Allocation mémoire non contiguë
- Insertion de valeur en tête

Schéma



et son code

```
class LinkedList<E>{  
    E data;  
    LinkedList<E> next;  
}
```

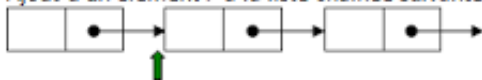
Recherche

Nécessairement séquentielle :

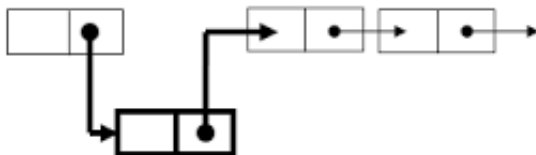
```
int search(int val, LinkedList theList){
    int index=0;
    for(LinkedList ll=theList;; !ll.isEmpty() \&\& ll.data
        ++index;
    }
    if(ll.isEmpty())
        return -1;
    return index;
    /* return ll.isEmpty() ? -1 : index; */
}
```

Insertion

Ajout d'un élément P à la liste chaînée suivante:

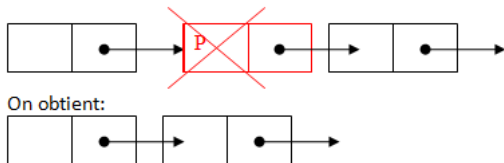


On obtient:



N.B. Faire attention à l'ajout en tête

Suppression



N.B. Faire attention à la suppression du premier élément

Bilan

Nous avons vu dans ce cours une SDD dynamique

Pour aller plus loin

- TDA piles/files ont une taille limitée à partir d'un tableau
- Une autre implémentation du TDA Liste :
l'ArrayList<E>