Éléments d'Algorithmique
CMTD6: les listes chainées

Christine Tasson
Université de Paris, IRIF

Listes chainées

Nous avons besoin d'une structure de données pour représenter une collection ordonnée d'objets du même type.

Exemple: 3, 0, -2, 5, 1, 4, -7, 0 est une liste d'entiers.

Pour le moment nous avons vu des tableaux.

Avantage : accès en temps constant à un élément du tableau.

Désavantage :

- Longueur du tableau fixée.
- Difficile de gérer l'élimination d'un élément.
- Difficile de gérer l'ajout d'un élément.

Listes chainées

Représentation d'une liste par une liste chaînée.

Liste chainée: structure de données de même type de taille arbitraire, dont la représentation en mémoire est une ensemble de cellules avec un contenu et un pointeur vers la cellule suivante

Avantages: Taille flexible, ajout et suppression en tète de liste

en temps constant.

D´désavantages: l'accès au i-ème ´élément de la liste n´nécessite de

parcourir les ´éléments qui le précèdent.

Recherche d’un élément dans un tableau

Pour aller plus vite, on peut utiliser les tableaux triés et la dichotomie, ou méthode "diviser pour régner".

Idée : si le tableau tab est trié, pour tout indice i,

- les éléments e 6 tab[i] sont d’indice 6 i,
- les éléments e > tab[i] sont d’indice > i.

On essaye avec i au milieu du tableau.

Insertion d’un élément x en tête de liste L

```
insertion_head(L: list, x: key){
        c= new cell()
        c.key=x
        c.next=L.head
        L.head=c
}
```

cellule=(c.key,c.next)
L.head=pointeur vers la première cellule de L.

Suppression d'un élément x en tête de liste L

```
remove_head(L: list){
        c=L.head
        if c != null {
                L.head = c.next
                }
        return c
}
```

cellule=(c.key,c.next)
L.head=pointeur vers la première cellule de L.